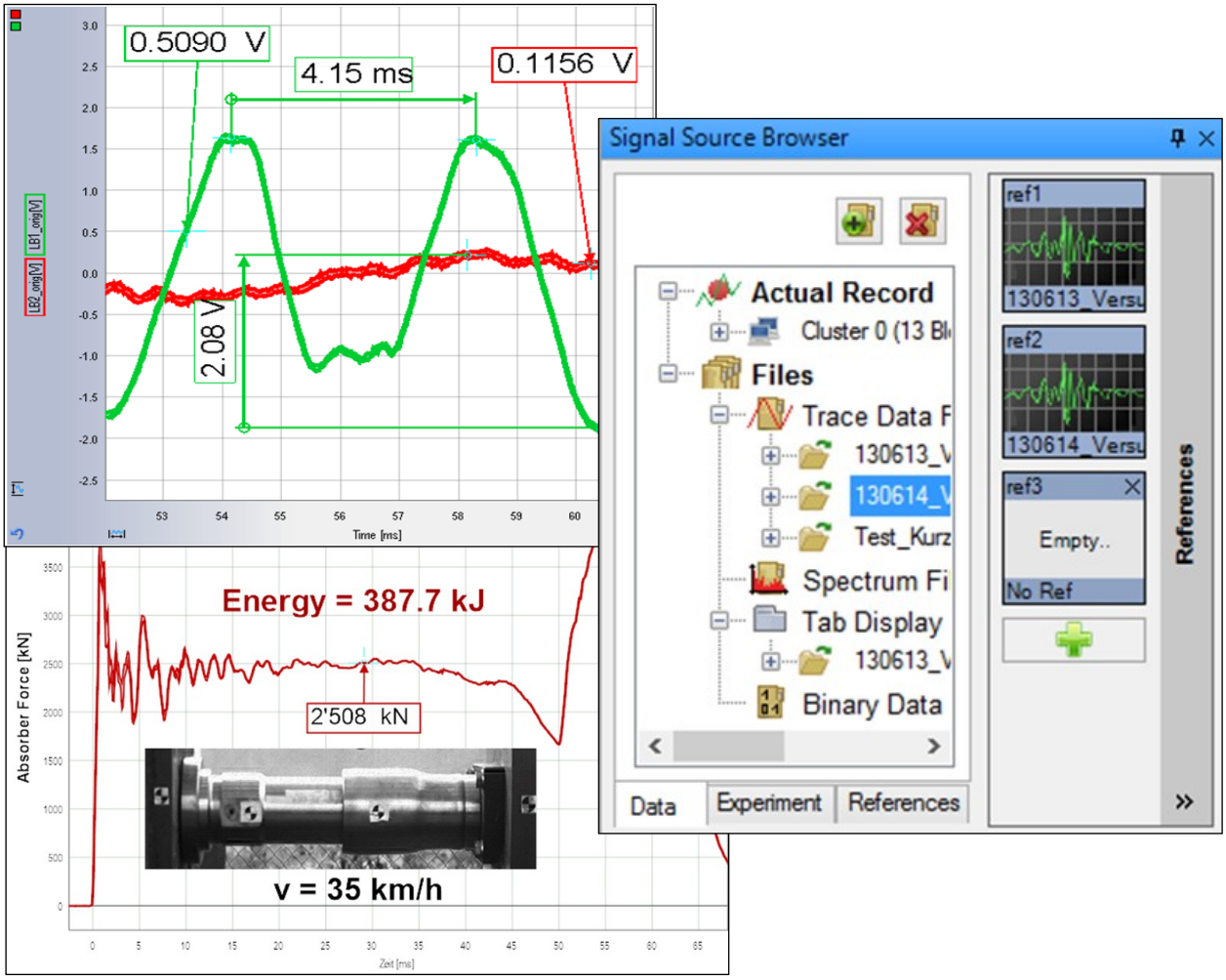
TranAX 3

Messdaten-Erfassung und Analyse



Benutzerhandbuch

Elsys AG
Mellingerstrasse 12
CH-5443 Niederrohrdorf

+41 56 496 01 55
www.elsys-instruments.com

Inhaltsverzeichnis

1 TranAX Übersicht 9
1.1 Anzeige & Funktionen 10
1.2 Icons 11
1.3 Anzeige von Experiment und Einstellungs-Datei 13
2 Erste Schritte 14
2.1 TPC Finder 14
2.1.1 TPC Device Finder: "Localhost" 15
2.2 Einfaches Experiment: FirstSteps 16
2.3 Eigene Aufnahme starten (Control-Panel Einstellungen dazu) 18
2.4 Aufnahmen speichern 20
2.5 Signale analysieren 21
3 SCOPE (Oszilloskop) 23
3.1 Kanal-Einstellungen 23
3.2 Knöpfe für Aufnahme-Befehle 24
3.3 Zeit-Einstellungen 24
3.4 Trigger-Bedingungen, Trigger-Pegel 25
3.5 Digital-Anzeigen 26
3.6 Maximale Kurven-Darstellung 26
4 Control Panel 27
4.1 Haupt-Einstellungen 28
4.2 Icons 29
4.3 Control Panel Register 29
5 Referenzen 30
5.1 Mögliche Quellen für Referenzen 30
5.2 Referenzen im Formeleditor 31
5.3 Referenzen in der Autosequenz 32
6 Aufnahme-Betriebsarten 33
6.1 Scope 33
6.2 Multiblock 33
6.3 Kontinuierlich 35
6.4 ECR Mode 35
6.4.1 Prinzipieller Ablauf 36
6.4.2 ECR Single Channel Mode 38
6.4.3 ECR Multi Channel Mode 38
6.4.4 Dual Mode 39
6.5 ECR Trigger Optionen 39

6.5.1 Normal 40
6.5.2 Holdoff 40
6.5.3 Retrigger 41
7 Eingangsverstärker 42
7.1 Mittelung (Averaging) 42
7.2 Verstärker-Einstellungen 42
7.3 Marker (Digital-Eingänge) 43
8 Trigger 44
8.1 Eingangs-Multiplizierer 45
8.2 AND Link (Logische UND Verknüpfung) 46
8.2.1 Beispiel 1: AND-Link (Zustand mit Flanke) 47
8.3 Pre- und Post-Trigger (Triggerverzögerung) 48
8.4 Trigger-Modi 49
8.4.1 Flanke 49
8.4.2 Fenster 49
8.4.3 Puls > Zeit 49
8.4.4 Puls < Zeit 49
8.4.5 Periode > Zeit 50
8.4.6 Periode < Zeit 50
8.4.7 Flankensteilheit 50
8.4.8 Zustand 51
8.5 Erweiterte Trigger-Funktionen (Übersicht) 51
8.5.1 Puls innerhalb t1 .. t2 52
8.5.2 Puls ausserhalb t1 .. t2 53
8.5.3 Verzögerung > t 54
8.5.4 Verzögerung < t 55
8.5.5 Verzögerung innerhalb t1 .. t2 56
8.5.6 Verzögerung ausserhalb t1 .. t2 57
8.5.7 Periode innerhalb t1 .. t2 58
8.5.8 Periode ausserhalb t1 .. t2 59
8.6 Bisherige Trigger-Modi bezogen auf Puls/Periode 60
8.6.1 Puls > t 60
8.6.2 Puls < t 60
8.6.3 Periode > t 61
8.6.4 Periode < t 61
9 Physikalische Einheit 62
9.1 Scale Designer 62
10 Informationsfenster 63

11 Cluster Konfiguration 64
12 Autosetup 65
13 Kurvenanzeige (Allgemeines) 69
13.1 Organisieren und Platzieren 69
13.2 Zoomen 70
13.3 Kurven verschieben 70
13.4 Volle Ansicht 70
14 Y/T-Kurven-Anzeige-Fenster 71
14.1 Legende 71
14.2 Text-Eintrag in Kurvenanzeige-Fenstern 73
14.3 Gitter 73
14.4 Hintergrundfarbe 73
14.5 Bereiche 74
14.6 Y-Achsen 74
14.6.1 Anzahl Y-Achsen links, rechts 74
14.6.2 Y-Achsen verriegeln 75
14.6.3 Y-Achsen Beschriftung 75
14.7 Kurvendarstellung 75
14.8 Cursor-Eigenschaften 76
14.9 Cursor auf Abtastpunkten 77
14.10 Vermessung von Kurven 78
14.11 Labels auf Kurven 79
14.12 Bilder einfügen, Resultate aus Berechnungen formatieren 80
14.13 Y-Skalierung von Kurven 81
14.14 Snapshot 81
14.15 Analyse von Multiblock-Aufnahmen (Block-Jumping) 82
15 Videos anzeigen, synchron zu Signal-Kurven 83
16 FFT Kurvenanzeige 85
16.1 Darstellung Skalierung 85
16.2 Oktaven- und Terz-Anzeigen 86
17 Aufnahmen Speichern & Drucken 87
17.1 Speichern 87
17.1.1 Speicherbereich 87
17.1.2 Datenreduktion 87
17.1.3 Kompression 87
17.1.4 Einstellungen hinzufügen 88
17.2 Exportieren 88
17.3 Drucken 88

 © Elsys AG

17.4 Zusätzliche Kurvenanzeigen 90
17.5 Speichern von Kurvenanzeige-Seiten 91
18 Signalquellen-Browser 92
18.1 HDF Viewer 93
18.2 Excel-Importer 93
19 Skalarfunktionen 94
19.1 Skalarfunktionen Tabelle A 95
19.1.1 Kanalauswahl zur Berechnung der Skalarfunktionen 95
19.1.2 Auswahl der Skalarfunktion 96
19.1.3 Beispiel 1: RMS 97
19.1.4 Beispiel 2: Frequency 98
19.1.5 Beispiel 3: Phase 100
19.2 Skalarfunktionen Tabelle B 102
19.2.1 Skalarfunktion hinzufügen 103
19.2.2 Beispiel 1: Apparent Power 105
19.2.3 Beispiel 2: Power Factor 106
19.3 FFT Funktionen (Tabelle A/B) 107
19.4 Zusätzliche Cursors 108
19.5 Auto-Refresh der Skalarfunktion Tabelle 109
19.6 Bedingte Hintergrundfarbe 109
20 Tabelle der Harmonischen 111
20.1 Erweiterte Harmonische Tabelle 114
21 Formeleditor 116
21.1 Beschreibung der Toolbar 117
21.2 Mit dem Formeleditor arbeiten 118
21.3 Cursors platzieren 120
21.4 String Variablen 120
21.5 Eigene Unterfunktionen definieren 120
21.6 Zahlenformat der Skalar-Resultate 122
21.7 Fehlermeldungen 122
21.8 Übersicht der Funktions-Gruppen 122
22 Mittelwertbildung über mehrere Aufnahmen 123
22.1 Mittelwertbildung im Zeitbereich 123
22.2 Mittelwertbildung von Spektren 124
23 Auto Sequenz 125
23.1 Auto Sequenzen in autonom laufenden TraNET-Geräten 126
24 Experimente 127
25 Weitere Menüpunkte 128

25.1 Benutzerspezifische Toolbars 128
25.2 Aufnahmeprotokoll (Menü "Ansicht") 128
25.3 Attribute (Menü "Ansicht") 129
25.4 Fehlerprotokoll (Menü "Ansicht") 129
25.5 Hardware Settings Viewer (Menü "Ansicht") 129
25.6 Sprache (Menü "Extras") 129
25.7 Einstellungen (Menü "Extras") 130
25.7.1 Import/Export (Menü "Extras") 130
25.7.2 Benutzeroberfläche (Menü "Extras") 131
25.7.3 Systemleistung (Menü "Extras") 133
25.7.4 Andere Einstellungen: Auto Save (Menü "Extras") 134
25.8 Kurven-Farben einem Kanal zuweisen (Menü "Extras") 134
25.9 Software Optionen Installieren (Menü "Extras") 136
25.10 Import-Option 137
26 Funktionsgruppen im Formeleditor 140
26.1 Gruppe "Channels" 140
26.2 Gruppe "All Functions" 140
26.3 Gruppe "Base Functions" 141
26.4 Gruppe "File Functions" 145
26.5 Gruppe "Signal Analysis" 151
26.6 Gruppe "Signal Processing" 154
26.7 Gruppe "Filter Functions" 158
26.8 Gruppe "Programming Functions" 159
26.9 Gruppe "Array Functions" 165
26.10 Gruppe "Exponential and Trigonometric" 167
26.11 Gruppe "Spectrum (FFT)" 168
26.12 Gruppe "Report Generator" 170
26.13 Gruppe "Recording Parameters" 174
26.14 Gruppe "Layout Waveforms" 175
26.15 Gruppe "Auto Sequence Functions" 176
26.16 Gruppe "Signal Generations" 179
26.17 Gruppe "Misc. Functions" 181
27 Liste der Auto Sequenz Befehle 184
28 Liste der Skalarfunktionen 189
28.1 All Functions 189
28.2 Vertical 190
28.3 Horizontal 196
28.4 Periodic 202

28.5 Cursor 203
28.6 Power 205
28.7 Misc 208
29 Diverses 213
29.1 ActiveX/COM - Interface 213
29.2 Sync.Clock Out 214
29.3 Konsolen Parameter 215
29.4 Verknüpfungen erstellen 217
29.5 Blockschaltbilder 218
29.6 Einschränkungen 220
29.6.1 Digitale Eingänge (Marker) 220
29.6.2 Differential-Eingänge 220
29.6.3 Maximale Abtastrate 220
30 Fehlerbehebung 221
30.1 TranAX Software Version 221
30.2 TPC-Server Version 221
30.3 Treiber und Firmware Version 222
30.3.1 Beispiel Windows XP 223
30.3.2 Beispiel Windows 7 223
30.3.3 TraNET FE 223
30.4 Fehlermeldungen 224
30.4.1 TranAX 224
30.4.2 TraNET Config Logfile 225

Vorwort

Betreiben Sie unsere **leistungsstarke Hardware**, TraNET, TPCX oder TPCE Module, mit der **Datenerfassungs- und Analyse-Software TranAX**. Dieses Benutzerhandbuch zeigt Ihnen alle TranAX Funktionen, Betriebsmodi und Einstellungen. Es gibt Ihnen einen Überblick über das System und dessen umfangreiche Fähigkeiten.

In TranAX taucht oft das Wort "**Experiment**" auf. Mit diesem **Begriff** ist eigentlich ein **Projekt** gemeint. Darin werden Einstellungen wie Messbereich, Abtastrate, Kanalnamen, die Anordnung der Fenster, Formeln, Autosequenzen usw. für einen bestimmten Messvorgang gespeichert. Ebenso werden normalerweise gespeicherte Signale innerhalb dieser Experiment-Umgebung verwaltet.

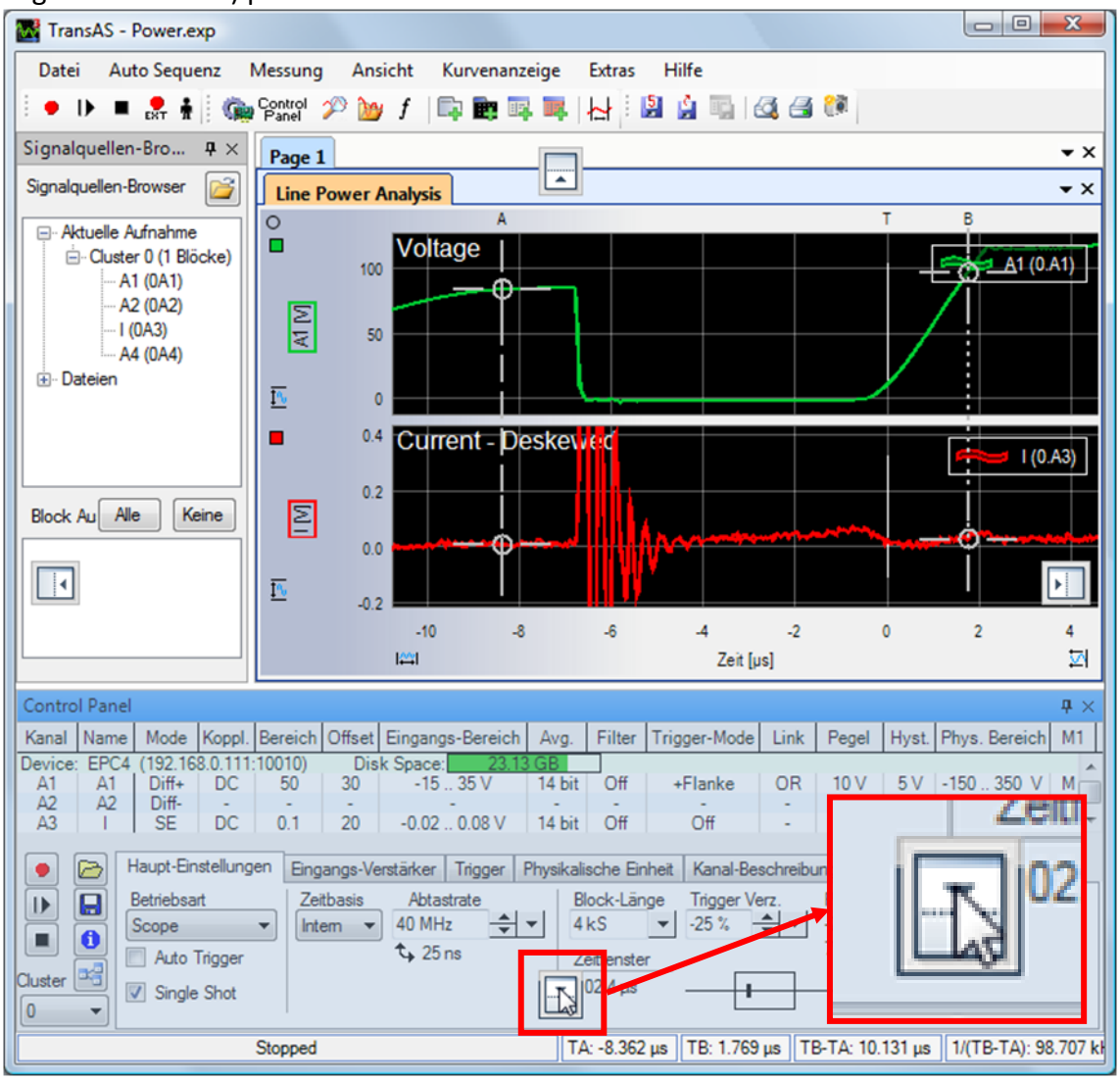
Falls TranAX für Sie neu ist, empfehlen wir Ihnen beim Kapitel Erste Schritte zu beginnen.

1 TranAX Übersicht

Das TranAX Messprogramm ist ein flexibel einsetzbares Werkzeug, um verschiedene Messaufgaben mit TraNET- und TPCX/TPCE-Datenerfassungs-Systemen durchzuführen. Es verwendet die geläufige **MDI-Schnittstelle** (Multi Document Interface) von Microsoft Windows.

Jede **Fenster-Art** wird im **Menü *"Ansicht"* aufgelistet.** Es kann jederzeit auf diese Liste zugegriffen werden. Es können mehrere verschiedene Fenster in einem Hauptfenster ("Seite" oder "Page") geöffnet werden. Einige dieser Fenster können zu einer Registerkarte minimiert am Hauptfenster angeheftet werden.

Um ein Fenster zu verschieben, kann es auf dem Registerkarten-Reiter oder am oberen Balken gepackt und auf die erscheinenden Platzierhilfen (Pfeile) gezogen und losgelassen werden. Dadurch wird es korrekt an der entsprechenden Seite (oder direkt überdeckend auf dem darunter liegenden Fenster) platziert.



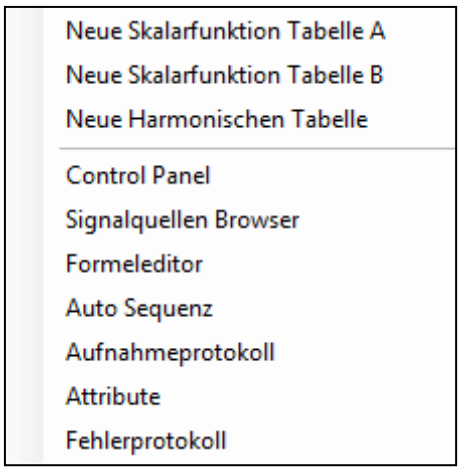
1.1 Anzeige & Funktionen

Die Oberfläche von TranAX besteht aus zwei Hauptfenstern: dem Control-Panel und der Kurvenanzeige. Das **Control-Panel** wird verwendet um alle **Hardware-Einstellungen** vorzunehmen. Es gilt zu beachten, dass geänderte Control-Panel-Einstellungen erst auf die nächste Aufnahme wirken.
Die **Kurvenanzeige** zeigt die Kurven von aufgenommenen oder laufenden **Signalen**. Es gibt verschiedene Anzeige-Fenster, welche im Menü ***"Ansicht"*** geöffnet werden können:

- *Neue Zoom Kurvenanzeige* um einen gezoomten Teilbereich von Signalen darstellen zu können.
- *Neue XY Kurvenanzeige* zeigt die XY-Ansicht der Signale.
- *Neue Marker Kurvenanzeige* um die digitalen Signale in einem separaten Fenster anzuzeigen.
- *Neue FFT Kurvenanzeige* zeigt das Frequenz-Spektrum des Signals an.

Jedes Fenster gehört zu einer *Seite*, welche ein Platzhalter für die Kurvendarstellungen und allfälligen Skalartabellen ist. Es muss immer zuerst mindestens eine *Neue Seite* geöffnet worden sein um Kurvenfenster anzuzeigen zu können.

Für weitere Funktionen können zusätzlich folgende Fenster im Menü ***"Ansicht"*** geöffnet werden:



- Öffnen einer neuen Skalartabelle A. Diese Tabelle eignet sich zur Berechnung von Kurvenparametern für mehrere Kurven.
- Öffnen einer neuen Skalartabelle B. Diese Tabelle eignet sich zur Berechnung von Kurvenparametern individuell pro Kanal.
- Die Tabelle der Harmonischen ermittelt die Grundwelle und die Harmonischen eines periodischen Signals.
- Der Signalquellen-Browser listet die aktuellen oder in Dateien gespeicherten Aufnahmen auf. Daraus können dann via Drag & Drop einzelne oder mehrere Kurven in die Anzeigefenster platziert werden.
- Mit dem Formeleditor können Signale berechnet und analysiert werden.
- Die Autosequenz ermöglicht eine Makro-Programmierung von Programmabläufen.
- Aufnahmeprotokoll: Im Continuous- oder ECR-Mode können Ereignis-Kommentare hinzugefügt werden. Darin werden auch automatisch alle Trigger-Ereignisse rapportiert. Einträge können auch im Nachhinein erfolgen.
- Attribute: Im Continuous- oder ECR-Mode können Kommentare oder zusätzliche Informationen (z.B. Versuchsnummer, Bedingungen, Name der Teilnehmer usw.) zu den Aufnahmen hinzugefügt werden.
- Fehlerprotokoll zeigt alle relevanten Programm-Ereignisse und Fehler.

 © Elsys AG

1.2 Icons

Die wichtigsten Icons sind in Gruppen auf der sog. Toolbar positioniert. Die meisten Funktionen können via folgenden Icons aufgerufen werden:

Control-Panel		
	Öffnet das Control-Panel	
	Öffnet den TPC Finder-Dialog, Gerät neu wählen	Wenn mehrere Geräte an einem Netzwerk angeschlossen sind, kann damit auf ein anderes Gerät verbunden werden.
	Öffnet das Auto Setup Fenster	Automatische Einstellung des Messbereichs, abhängig von der anliegenden Signalamplitude
	Öffnet den Signalquellen-Browser	Ermöglicht den Zugriff auf Signale, auch aus Dateien
Recording Commands		
	Aufzeichnung starten (F6)	
	Manueller Trigger (F7)	
	Aufzeichnung manuell beenden (F8)	
	Gibt die Möglichkeit frei, via externes TTL-Signal eine Messung zu starten	Stecker-Anschluss siehe Hardware-Handbuch
	Status-Anzeige der Datenerfassung	Animation (Ergänzend zur gelben Anzeige links unten)
	Mittelwertbildung	Mittelwertbildung aus mehreren Aufnahmen (im Scope-Mode)
Layout Control		
	Öffnet ein neues SCOPE-Fenster	
	Öffnet eine neue Seite.	
	Fügt eine neue Kurvenanzeige hinzu	
	Fügt eine neue FFT-Kurvenanzeige hinzu	Default-Position siehe unter Menü "Extras"
	Fügt eine Skalar Funktions-Tabelle A hinzu	Default-Position siehe unter Menü "Extras"
	Fügt eine Skalar Funktions-Tabelle B hinzu	Default-Position siehe unter Menü "Extras"
	Verknüpft das Cursor-Paar in der Kurvenanzeige	Dadurch werden die Cursor (in der Regel A und B) gemeinsam verschoben
	Öffnet den Formeleditor	
Save Options		
	Speichert alle Einstellungen	Hardware, Layout, Formeln, Autosequenz
	Lädt alle Einstellungen	
	Speichert Signalkurven (Zeitbereich) im TPC5 Format	Das Datei-Format entspricht dem HDF5-Standard)

	Speichert Spektrumkurven (Frequenzbereich) im TPS5 Format	Die Spektrum-Kurven müssen vorher in einem FFT-Fenster erzeugt worden sein
	Aufzeichnung exportieren (als .TPC-, DI-Adem- oder ASCII-Format usw.)	TPC-Dateien lassen sich in TranAX Version 2 öffnen
	Skalar Funktionstabelle in eine ASCII-Datei exportieren	Erzeugt eine Text-Datei von der aktuellen (fokussierten) Skalar-Tabelle
	Speichert eine komplette Seite	Dabei können die Quellen-Referenzen der angezeigten Kurven substituiert werden, sodass sie danach auf die entsprechenden Kurven in der Datei zeigen.
	Lädt eine komplette Seite	
	Zeigt die Druckvorschau.	Hier lässt sich auch der Ausdruck konfigurieren
	Druckt die via Druckvorschau konfigurierte Kurvenanzeige	
	Speichert die aktuelle Kurvenanzeige in die Windows Zwischenablage.	Unter Menü "Extras" können dazu verschiedene Parameter eingestellt werden. Via das Dreieck können ganze Seiten (inkl. Skalar-Tabellen usw.) in die Zwischenablage kopiert werden.
Block Jumping		Werden vor allem bei Multiblock- und ECR-Aufnahmen verwendet
	Vorheriger Block	Zeitfenster springt zum vorherigen Block
	Nächster Block	Zeitfenster springt zum nächstfolgenden Block
	Arretiere Cursor auf dem Bildschirm	Die Cursor bleiben stehen, auch beim Zoomen und Verschieben der Kurven (sie sind nicht an die Kurven gekoppelt)
	Arretiere Zeitfenster auf dem Bildschirm	Die Zeitfenster-Marken im Hauptfenster bleiben stehen, auch beim Zoomen und Verschieben der Kurven (sie sind nicht an die Kurven gekoppelt)
	Setze Zeitfenster auf Block	Setzt die Zeitfenstermarken auf die Grenzen eines Blocks

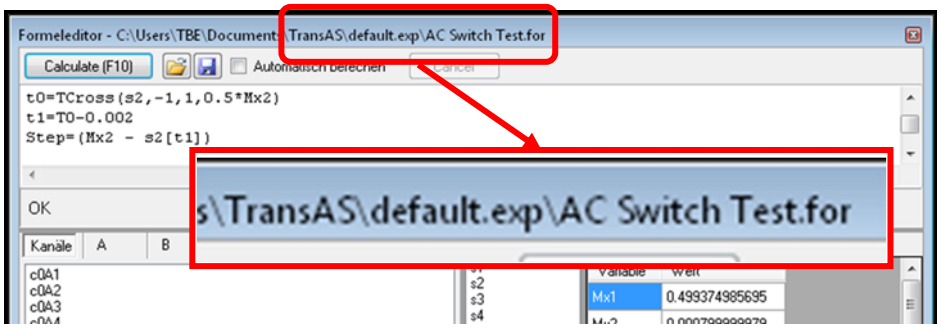
© Elsys AG

Diverse Icons		
	Die letzten Änderungen in der Kurvenanzeige rückgängig machen (Undo)	Es betrifft dies nur das Zoomen und Verschieben von Kurven
	Die letzten Änderungen in der Kurvenanzeige wiederherstellen (Redo)	Es betrifft dies nur das Zoomen und Verschieben von Kurven
	Pin Option um Fenster als Registerkarte zu minimieren	

1.3 Anzeige von Experiment und Einstellungs-Datei

Im oberen Balken des TranAX-Fensters wird der Name inklusive Pfad des geladenen Experiments sowie der aktuellen Layout-Einstellung angezeigt. Das gleiche gilt für die Autosequenz- und Formel-Editor-Fenster.

Man kann so erkennen, woher die Einstellungen ursprünglich übernommen worden sind.



2 Erste Schritte

Dieses Kapitel gibt eine Einführung in die fast unbegrenzten Aufnahme- und Auswerte-Möglichkeiten von TranAX.

Für Anwendungen, bei welchen auf die speziellen Möglichkeiten verzichtet werden kann, verweisen wir auf die sogenannte SCOPE-Anwendung.

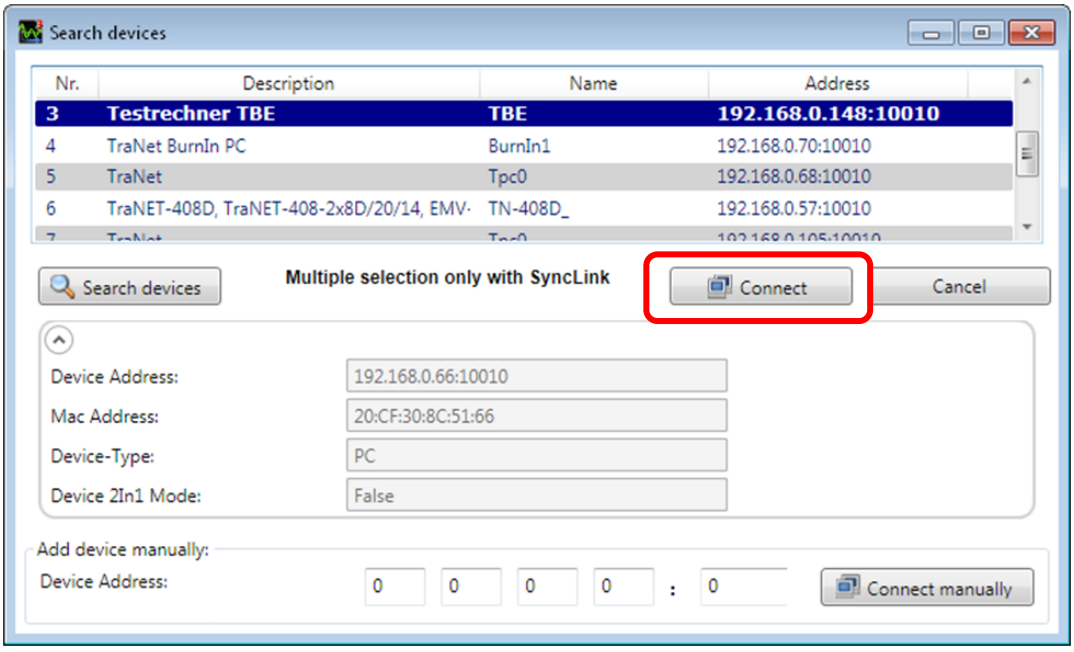
Unter "Erste Schritte" wird gezeigt, wie man von zwei Kanälen Signale aufzeichnet, analysiert und speichert. Es werden die grundlegenden Arbeitsweisen und Arbeitsschritte beschrieben.

Zuerst werden wir ein einfaches vorbereitetes Experiment durchführen und anschliessend Schritt für Schritt die wichtigsten Funktionen von TranAX kennenlernen.

2.1 TPC Finder

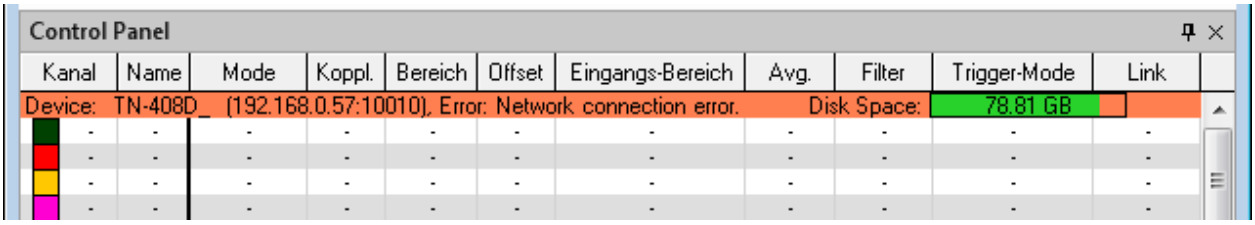
Um die eingebauten TPCX/TPCE-Module oder externe Geräte wie TraNET FE zu benutzen, muss TranAX mit diesen verbunden werden. Da die gesamte Kommunikation TCP/IP basierend ist, werden diese, wie in einem Netzwerk üblich, über deren IP-Adresse und dem entsprechenden Port angesprochen.

Beim ersten Start von TranAX oder über das Menü ***"Datei" / "Gerät neu wählen.."*** wird der TPC Finder (Search Devices) gestartet:



Die zurzeit aktive Verbindung wird dunkelblau dargestellt, die anderen Geräte befinden sich ebenfalls im Netzwerk, sind aber nicht verbunden.

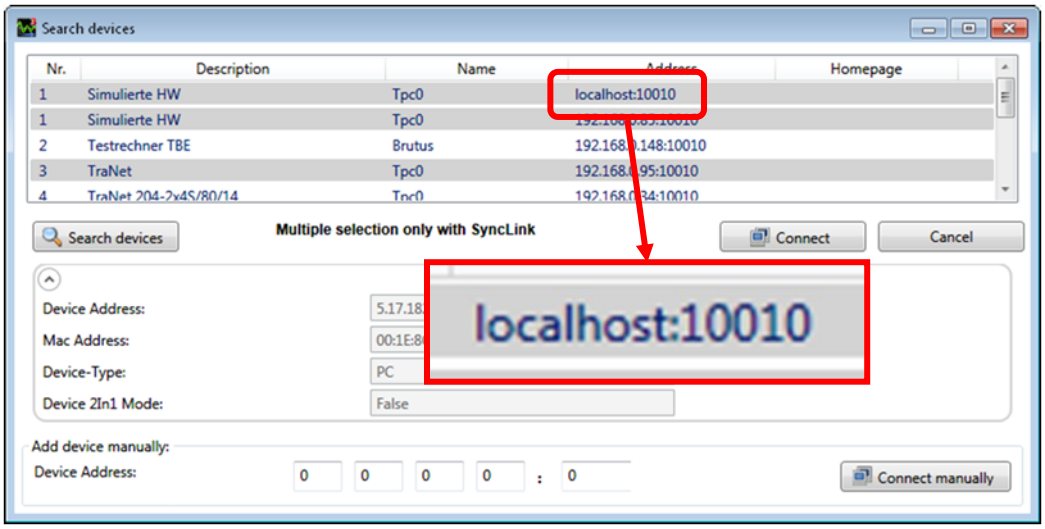
Fall Sie mit einem anderen Gerät verbinden möchten, wählen Sie dieses an und klicken Sie den Knopf "Connect", anschliessend wird TranAX neu gestartet und die Verbindung zu dem Gerät steht. Falls Sie das Gerät vom Netz nehmen wird dies im Control-Panel in oranger Farbe signalisiert.



2.1.1 TPC Device Finder: "Localhost"

Die **lokalen TPCX/TPCE-Module** (TraNET-PPC oder TraNET-EPC) werden im TPC-Finder nicht mit der IP-Adresse des Rechners verknüpft, sondern **direkt als "localhost" abgelegt**.

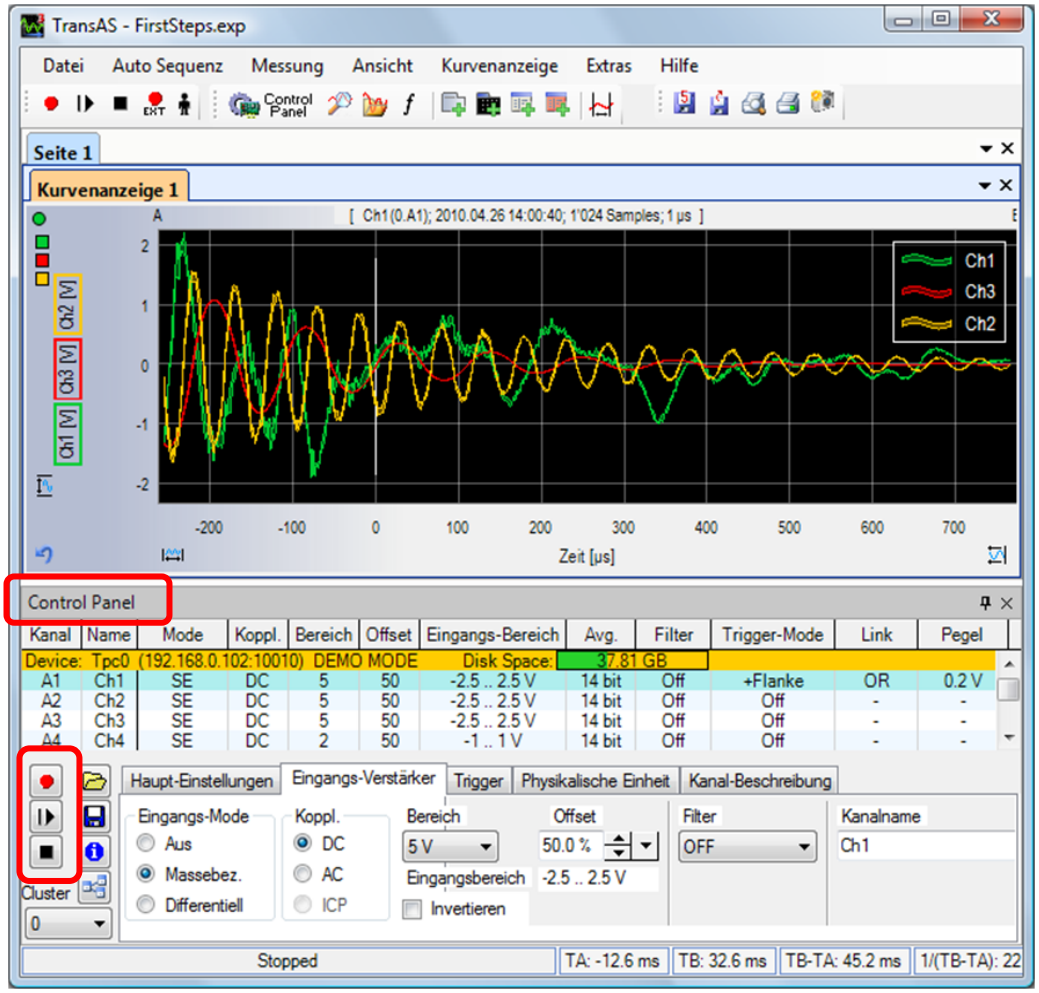
Dadurch verbindet sich TranAX, unabhängig vom Status der Netzwerkverbindung oder der Art des Netzwerks (Firmennetz, Kundennetz oder keine Netzwerkverbindung), nach dem Start direkt mit den lokalen Modulen. Dies hat den Vorteil, dass die Verbindung ohne allfälligen Umweg über ein (zufällig) vorhandenes Netzwerk erfolgt.



"Localhost" verweist auf die interne IP-Adresse eines Rechners welche immer **127.0.0.1** ist. Auch wenn keine Netzwerk-Hardware in dem Rechner installiert ist, lässt sich diese Adresse ansprechen und ist **unabhängig** von den Netzwerkeinstellungen.

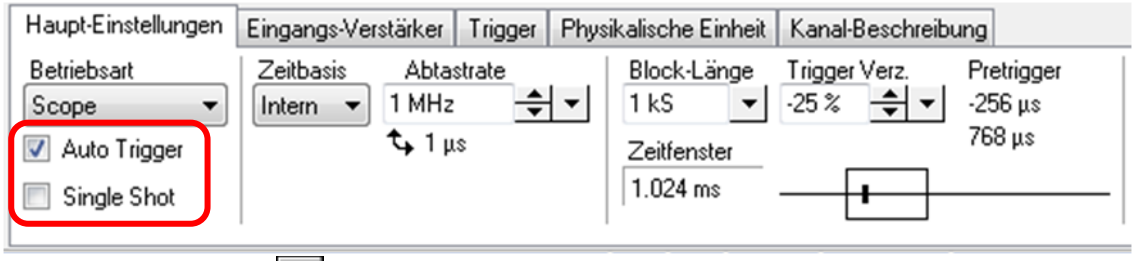
2.2 Einfaches Experiment: FirstSteps

Laden Sie das Experiment "FirstSteps" via ***"Datei" / "Experiment öffnen"***. Die Experimente liegen normalerweise im Standardverzeichnis des Computer-Benutzers *Eigene Dateien\TranAX* Je nach Installation und verwendeter Windows-Version können diese Ordner anders heissen.
Danach Laden Sie via "Datei / "Alle Einstellungen laden" die Einstellungen *"FirstSteps_Scope1.lay".*



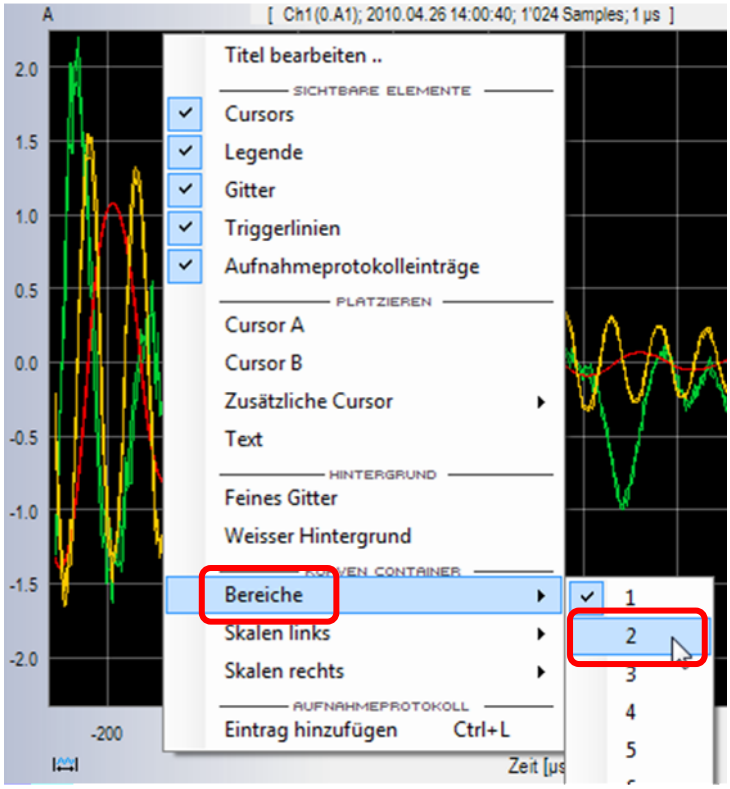
Durch Anschliessen von einfachen Drähten an den BNC-Buchsen A1 und A3 können EMI-Signale ("Brumm-Signale") gemessen werden. Sie können auch andere Signale anschliessen, wie z.B. einen Funktionsgenerator.

Durch Klicken auf den roten **Start-Knopf** wird das Gerät in den Datenaufzeichnungs-Mode geschaltet. Da im Control-Panel **Auto Trigger** aktiviert und **Single Shot** deaktiviert ist, sollten jetzt im Kurvenanzeige Fenster laufend neue Kurven angezeigt werden.



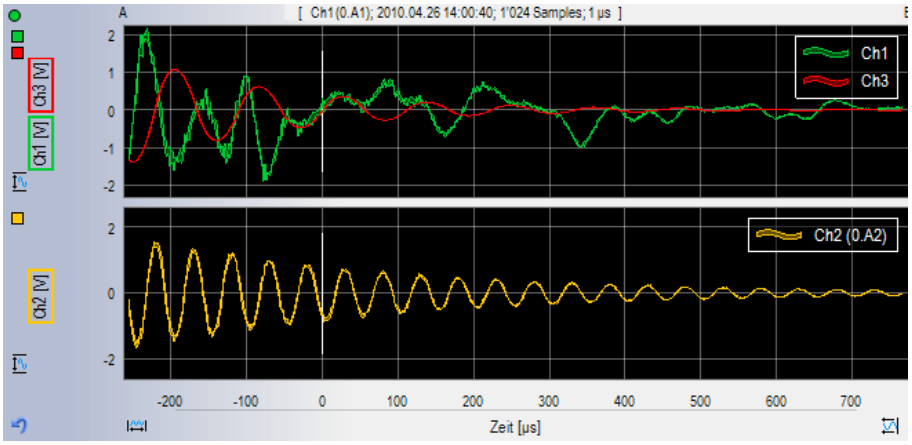
Mit dem Stop-Knopf wird die Aufnahme abgebrochen.

 © Elsys AG



Um jeden Kanal separat in einer Kurvenanzeige darstellen zu lassen, klicken Sie mit der **rechten Maustaste** auf den Kurven-Hintergrund in der Kurvenanzeige und wählen Sie im Kontextmenü *Anzahl* ***Bereiche auf 2***.

Verschieben Sie die zweite Signal Kurve von der oberen in die untere Anzeige mittels Drag & Drop. Platzieren Sie dazu die Maus über dem farbigen Kanal-Kästchen (links neben der Kurvenanzeige) und ziehen Sie das Kästchen auf die untere Anzeige.



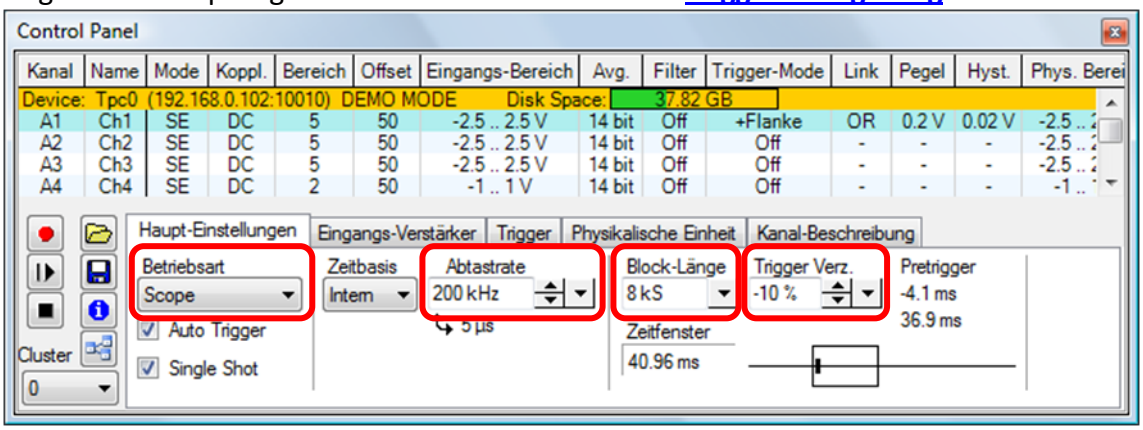
2.3 Eigene Aufnahme starten (Control-Panel Einstellungen dazu)

Bevor wir eine neue Aufnahme starten, werden wir einige Hardware Parameter (Abtastrate, Blocklänge, Zeitbasis, usw.) mit Hilfe des Control-Panels ändern.
Klicken Sie im Menü ***"Ansicht" / "Control Panel"*** oder drücken Sie einfach das Control Panel Icon

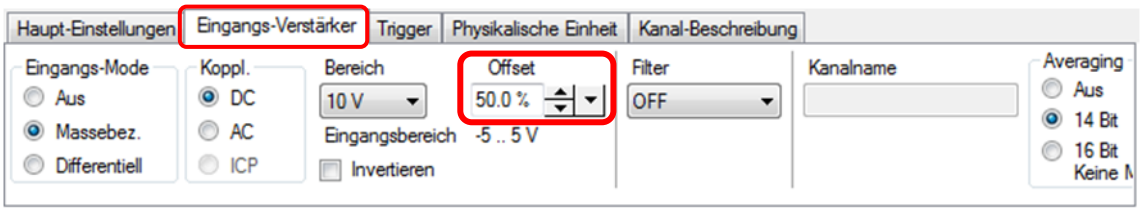
. Markieren Sie die ersten drei Kanäle in der Liste (A1 bis A3) um die Einstellungen dieser Kanäle gleichzeitig zu ändern. Jeder Listeneintrag entspricht einem Kanal des TPCX/TPCE Modules.

Haupt- und Zeitbasis Einstellungen: Stellen Sie die Betriebsart auf Scope Mode. Für mehr Informationen über die Betriebsmodi Einstellungen schauen Sie sich den Abschnitt Haupt-Einstellungen an. Und falls Sie mehr über die Funktionsweise dieser Modi erfahren möchten, gehen Sie zum Abschnitt Aufnahmeprinzipien.

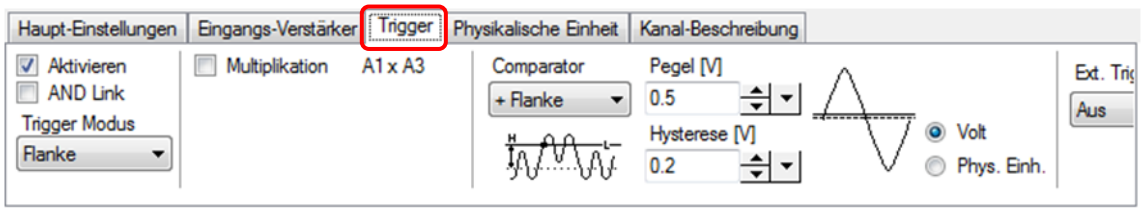
Geben Sie im **Abtastrate**-Textfeld 200 kHz ein oder verwenden Sie dazu die Dropdown Liste. Für die Blocklänge stellen wir 8 kS (kilo Samples) ein. Unter der Eingabe **Block-Länge** sehen wir das resultierende Aufnahme-**Zeitfenster**. In unserem Beispiel haben wir die Abtastrate verdoppelt und die Aufzeichnungslänge halbiert. D.h. das resultierende Zeitfenster ist nur noch ein Viertel so gross wie ursprünglich. Schliesslich setzen wir die **Triggerverzögerung** auf -10%.



Auf der Eingangs-Verstärker Registerkarte ändern wir nur den Bereich (10V) und belassen den Offset bei 50%.

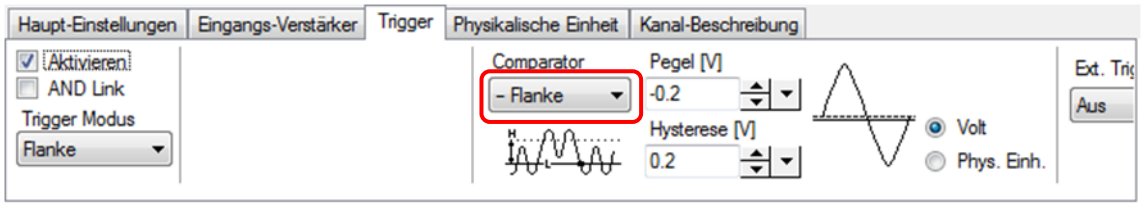


Als nächstes stellen wir die **Trigger**-Einstellungen ein. Wir wählen **Flanke** als **Trigger-Modus**, d.h. wir wollen, dass auf positive Flanken bei einem Pegel grösser als +0.5V getriggert wird. Wir setzen den **Pegel** auf 0.5V und die **Hysterese** auf 0.2V. Ausserdem muss die Triggerquelle aktiviert sein, bevor ein Trigger ausgelöst werden kann. Markieren Sie dazu die Schaltfläche **Aktivieren**.

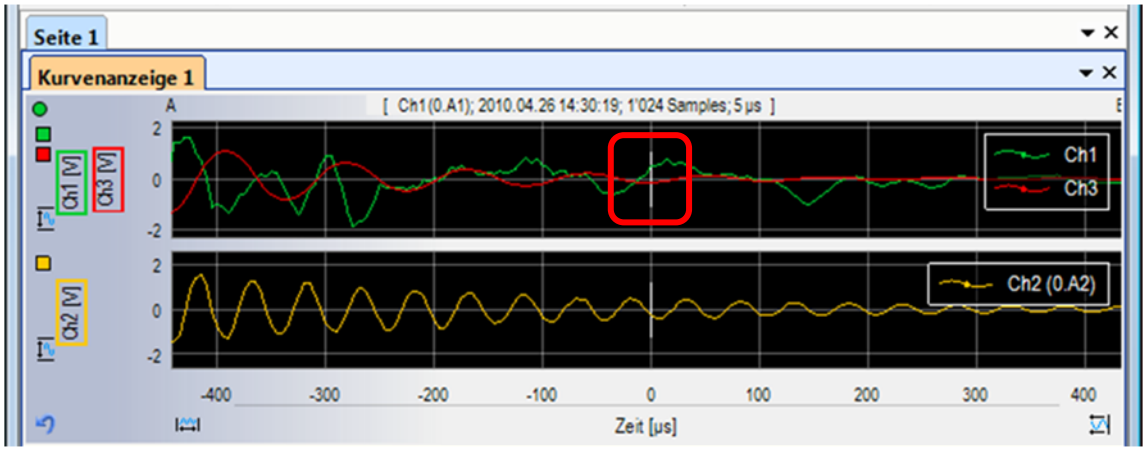


Falls eine andere Einheit als Volt gewünscht wird, gehen Sie in die Registerkarte **Physikalische Einheit**. In unserem Fall arbeiten wir in Volt und belassen die Einstellungen dort.

Nun haben wir die Kanäle A1 bis A3 alle gleich eingestellt. Später wollen wir, dass Kanal A1 auf die positiven Flanken triggert und Kanal A3 auf die negativen Flanken. Wählen Sie dazu nur Kanal A3 im Control Panel aus und gehen Sie zu der Registerkarte **Trigger**. Wechseln Sie den **Komparator** zu **-Flanke**, den **Pegel** auf -0.2V und die **Hysterese** auf 0.2V.



Wir sind bereit eine neue Aufnahme zu starten. Um die Aufnahme zu starten, drücken Sie den **Start-Knopf** oder die F6-Taste auf der Tastatur.
TranAX zeigt nun die Meldung **"Armed"** in der Statusanzeige unten links vom Fenster an. Schliessen Sie ein steigendes Signal (>0.5V) beim Eingang A1 an. Sobald ein Trigger ausgelöst wurde, wird die Aufnahme beendet und die Status Anzeige wird **"Stopped"** anzeigen. In der oberen Kurvenanzeige sehen Sie nun das aufgenommene Signal mit einer steigenden Flanke über 0.5V.

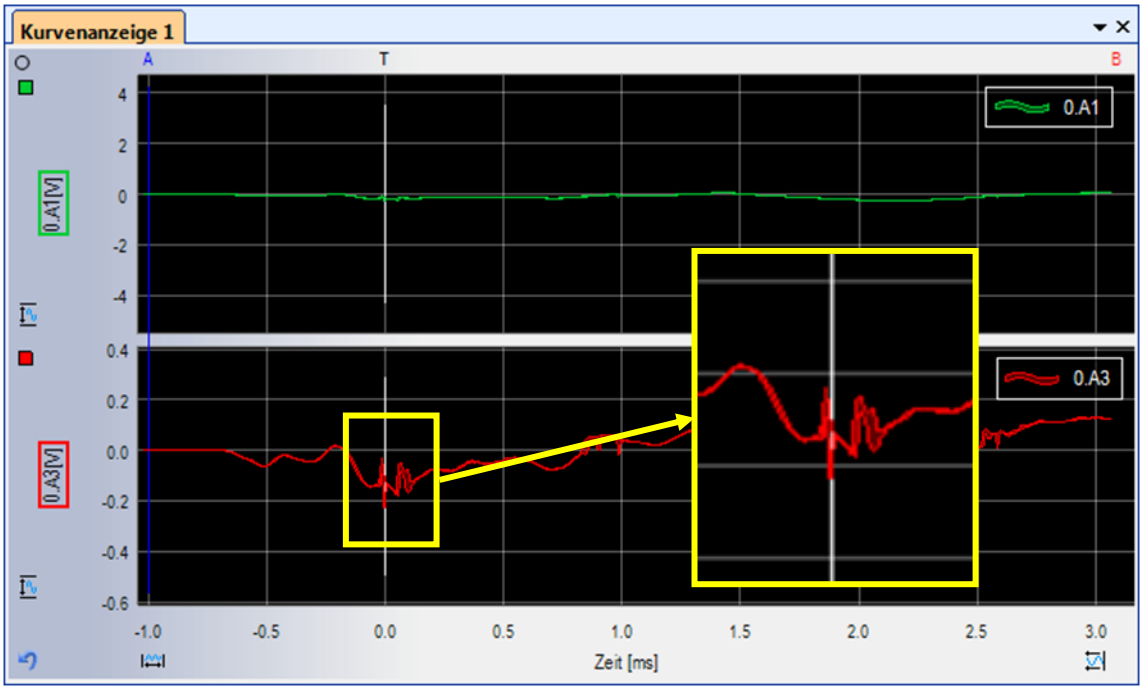


Die Kurven sind eventuell nicht optimal in der Anzeige platziert weil wir einige Aufnahmeparamater geändert haben seit der letzten Aufnahme. Um die Kurve vollständig anzeigen zu lassen, klicken Sie auf die folgenden Symbole:

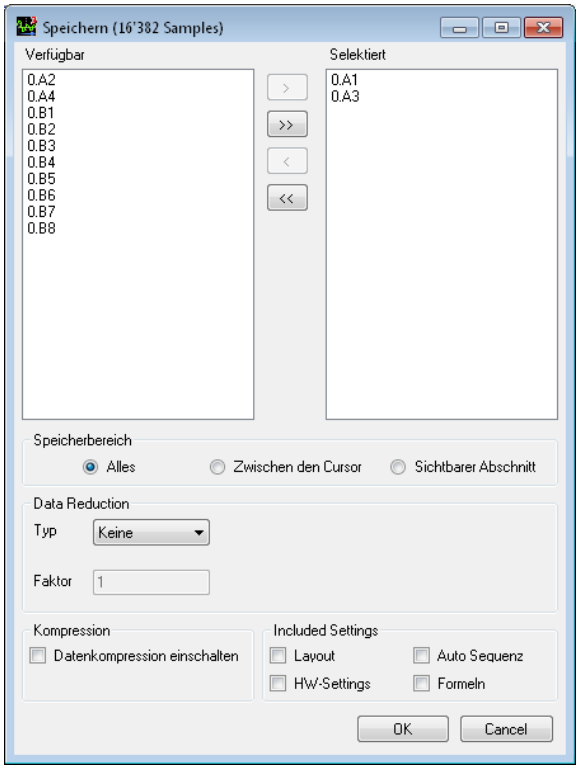
	Volle Ansicht Y
	Volle Ansicht X

Anschliessend kann beliebig gezoomt werden.

Entfernen Sie wieder den Anschluss am Eingang A1 und schliessen Sie eine Signalquelle mit fallender Flanke (< -0.2V) am Eingang A3 an. Starten Sie die Aufnahme. Sie sollten anschliessend in der unteren Anzeige das aufgenommene Signal mit fallender Flanke sehen.



2.4 Aufnahmen speichern



Nachdem wir die Aufnahmen beendet haben, werden wir die zwei Kurven Signale speichern. Um die aktuellen Aufnahmen zu speichern, wählen Sie im Menü "***Datei" / "Speichern"*** *als* ***tcp5-Datei*** oder klicken Sie auf das Icon . Wählen Sie aus der linken Liste *Verfügbar* die Kanäle 1 und 3 aus und fügen Sie sie der Liste *Ausgewählt* mittels der Pfeilschaltfläche hinzu. Belassen Sie die anderen Optionen wie sie sind. Für mehr Informationen über die Speicher Optionen, gehen Sie zum Abschnitt Aufnahmen speichern & drucken.

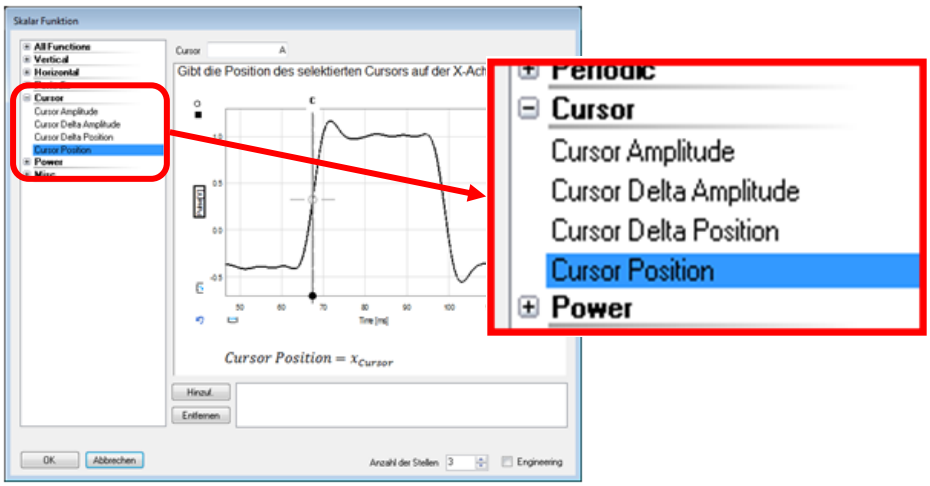
Da wir mit der Aufnahme fertig sind, brauchen wir das Control Panel nicht mehr und minimieren es mit der Pin-Option zu einer Registerkarte, die danach am unteren Rand links zu finden ist.

2.5 Signale analysieren

Um die aufgenommen Signale zu analysieren, verwenden wir die Skalar-Funktionen. In diesem Beispiel wollen wir den Spitzenwert der Signale finden und setzen die Cursor A und B in der Kurvenanzeige entsprechend. Verschieben Sie den Cursor B etwa ans Ende der Aufnahme und den Cursor A nach links bzw. an den Anfang der Aufnahme.

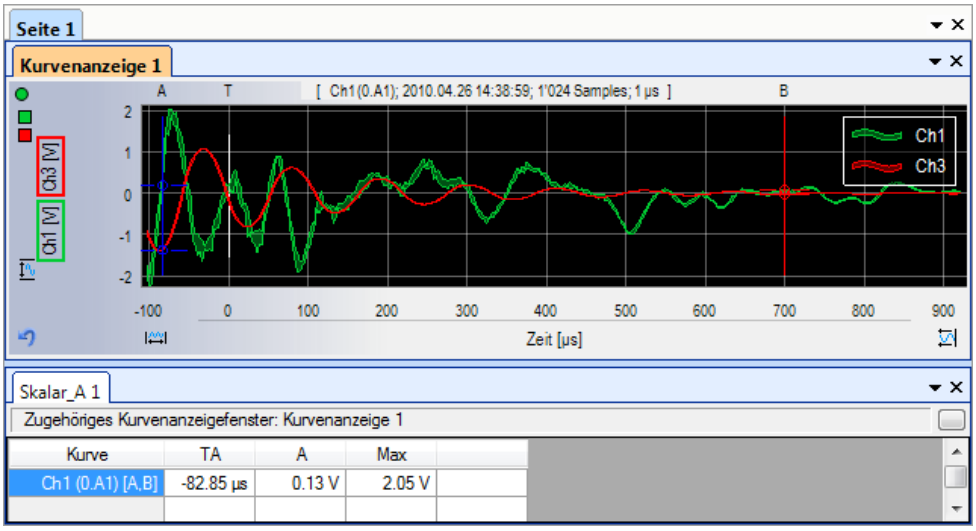
Wählen Sie "Kurvenanzeige 1" als das aktive Fenster und drücken Sie den Knopf ***"Neue Skalartabelle A"*** . Eine Registerkarte "Skalar_A 1" erscheint rechts neben *Kurvenanzeige 1*.

Wir werden nun drei Skalarfunktionen verwenden um die Signale zu analysieren. Zuerst wollen wir die Position von Cursor A wissen. Rechtsklicken Sie auf die erste Kolonne und wählen Sie *Skalarfunktion...*. Im Dialogfenster wählen Sie dann Gruppe *Cursor/Cursor Position*.

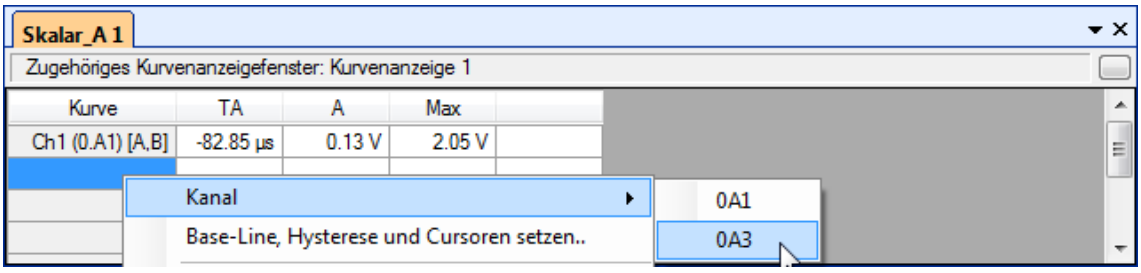


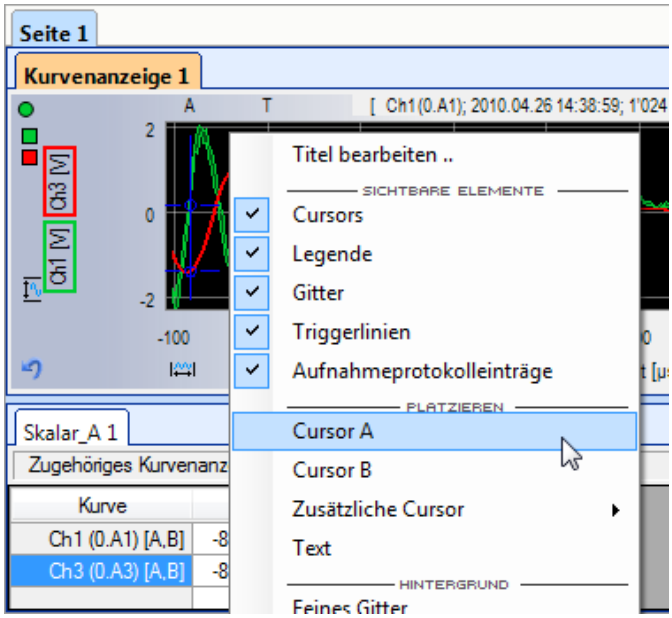
Wiederholen Sie die Schritte von oben für ***Cursor Amplitude*** und schliesslich fügen Sie noch ***Maximum*** aus der Gruppe ***Vertikal*** hinzu.

Die Kolonne *TA* (Time of Cursor A) zeigt uns die Position von Cursor A, die Kolonne *A* den Wert an der Stelle des Cursor und *Max* (3. Kolonne) zeigt das gefundene Maximum zwischen Cursor A und B.



Um das Signal von Kanal 3 zu analysieren, klicken Sie mit der rechten Maustaste auf eine freie Zeile in der Kolonne *Kurve* und wählen Sie *Kanal/0A3*.





Nun müssen Sie noch die Cursor A und B an die gewünschten Stellen verschieben. Falls die Cursor ausserhalb des Fensters sind, klicken Sie mit der **rechten Maustaste** in die Kurvenanzeige und wählen Sie *Cursor A oder Cursor B* Platzieren. Beachten Sie, dass Sie dabei den Mauszeiger an die Stelle positionieren, an der Sie den Cursor gerne hätten.

Bereits gespeicherte Signale können wieder geladen und analysiert werden. Das Signal einer TPC5-Datei kann mit dem Signalquellen-Browser geöffnet werden. Drücken Sie den Knopf .

Öffnen Sie nun im Signalquellen-Browser mit dem Knopf oben rechts die Datei-Dialog-Maske und navigieren ins Verzeichnis ***FirstSteps.exp/data***.

Um die gespeicherten Signale in einem separaten Display anzeigen zu lassen, klicken Sie mit der rechten Maustaste in das Kurvenanzeige Fenster und wählen Sie *Bereiche/3*. Mittels Drag & Drop kann man nun das gewünschte Signal vom Signalquellen Browser in das neue Kurven Fenster ziehen. Passen Sie darauf die Anzeige dem Signal noch mit dem Knopf *Volle Ansicht X* an.

Vielleicht haben Sie bemerkt, dass die gespeicherten Signale von der Datei grössere Zeitbereiche allozieren, weil wir die relevanten Aufnahme-Parameter für die neue Aufnahme geändert haben. TranAX kann mit solchen unterschiedlichen Aufnahmen ohne Einschränkungen umgehen. Sie können nun wie oben beschrieben die neu hinzugefügten Signale mit den Skalarfunktionen ausmessen.

3 SCOPE (Oszilloskop)

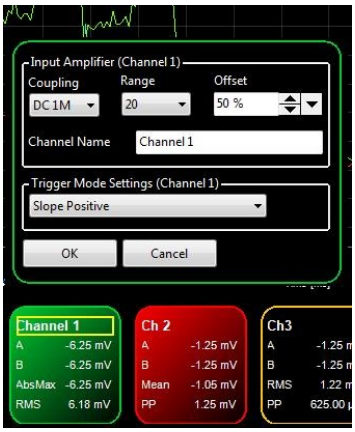
Mit SCOPE wird das Arbeiten bei einfacheren Anwendungen wesentlich erleichtert. TranAX verhält sich dabei wie ein Oszilloskop. Zwar immer noch ohne Drehknöpfe, aber für die Bedienung wurden selten gebrauchte Modes und deren Einstellungen stark in den Hintergrund gebracht.
Alle Manipulationen, welche die Kurven, deren Fenster, Achsen, Beschriftungen usw. betreffen verhalten sich gleich, wie bei einem normale Y/T-Fenster.

Durch Klick auf das Icon rd ein SCOPE-Fenster erzeugt.

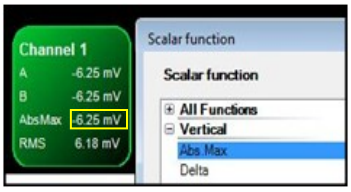
3.1 Kanal-Einstellungen

Im unteren Teil befindet sich für jeden Hardware-Kanal eine Bedienfläche.
Je nach Hardware sind 4 oder 8 solcher Bedienflächen vorhanden (bei Systemen mit mehr als 8 Kanälen, werden nur die ersten acht berücksichtigt).
Links befindet sich die Bedienfläche für den ersten Kanal (BNC A1). Die Bedienflächen sind vor allem durch die Namen der Kanäle gekennzeichnet.

Normalerweise wird ein gelber Rahmen um die ganze Bedienfläche gezeichnet, wenn sich der Maus-Cursor innerhalb der Fläche befindet. Durch Drücken der linken Maus-Taste kann der Kanal (inkl. Kurve) aktiviert oder deaktiviert werden. Durch das Aktivieren wird normalerweise die Kurve sichtbar.



Wenn sich der Maus-Cursor über einem Kanal-Namen befindet, wird dieser mit einem gelben Rahmen versehen. Durch Links-Klick öffnet sich eine Maske zur Einstellung der wichtigsten Kanal-Parameter.



Ein gelber Rahmen wird auch gezeichnet, wenn der Maus-Cursor über einem Skalarwerte-Bereich platziert wird. Durch Klick mit der linken Maus-Taste öffnet sich die Maske für die Auswahl und Spezifizierung einer Skalar-Funktion.

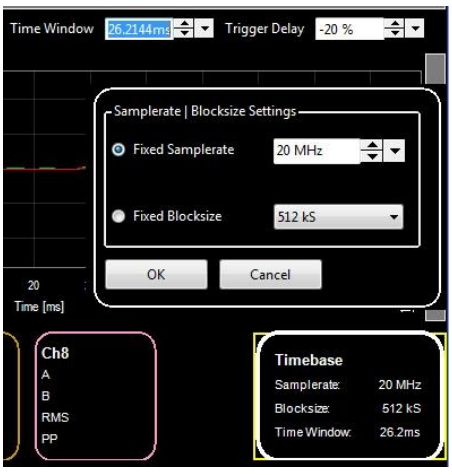
3.2 Knöpfe für Aufnahme-Befehle

Im oberen Bereich befinden sich die Befehlsknöpfe. Wie erwähnt kann damit wie mit einem Oszilloskop gearbeitet werden. Weitergehende Aufnahme-Modes (z.B. Multi-Block, ECR usw.) werden nicht unmittelbar ermöglicht.

Auto: Mehrfachauslösung mit "Auto-Trigger" (freilaufendes Oszilloskop).
Normal: Mehrfachauslösung, wartet auf Signal-Trigger und startet neu nach Abschluss der Aufnahme.
Single: Startet Aufnahme und wartet auf Trigger.
Nach Abschluss der Aufnahme wird gewartet.
Trigger : Das Programm schickt einen Trigger zur Hardware. Damit kann bei Fehlen eines Signal-Triggers eine laufende Aufnahme richtig abgeschlossen werden.
Stop: Eine laufende Aufnahme wird abgebrochen. Danach wird gewartet.

3.3 Zeit-Einstellungen

Mit dem **Time Window** Parameter kann der Zeit-Bereich eingestellt werden. Nach einer Aufnahme entspricht dies normalerweise dem Bereich bei voller X-Achse (**Knopf** **links unten**).



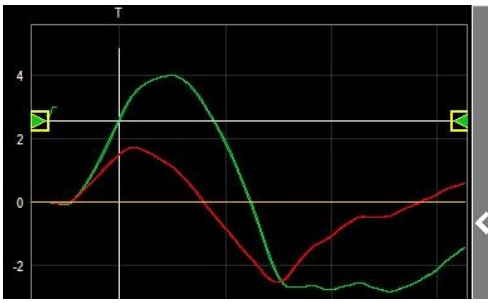
Der Zeit-Bereich errechnet sich wie folgt:
*T = Blocklänge * (1/Abtastfrequenz)*.
Der Anwender hat nun die Möglichkeit zu wählen, welche der beiden Termen vorgegeben und dann konstant sein soll. Durch Klicken auf dem **Timebase**-Feld rechts unten öffnet sich eine Maske. Darin kann zwischen "fixer Abtast-Frequenz" oder "fixer Block-Länge" und dem dazugehörenden Werte gewählt werden. Der andere (nicht fixierte) Wert wird danach automatisch aus dem unter **Time Window** gewählten Zeit-Fenster ermittelt.

 © Elsys AG

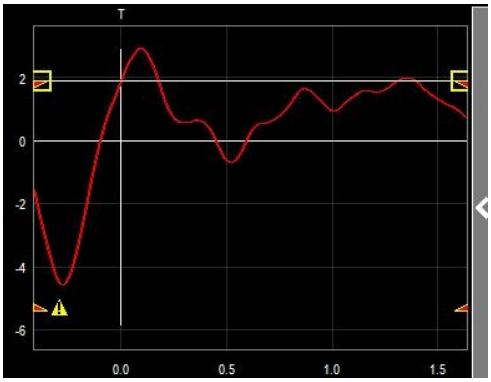
Unter **Trigger Delay** wird die Trigger-Verzögerung (-100% bis +200%) eingestellt. Diese Einstellung beeinflusst ebenfalls die Grenzen der Zeitachse. Allenfalls muss auch via
auf vollen Bereich der X-Achse gestellt werden.

3.4 Trigger-Bedingungen, Trigger-Pegel

Die Trigger-Bedingungen werden auch in der Maske der Kanal-Einstellungen gesetzt.
Sie sind bewusst einfach gehalten. D.h. es kann hier nur Flanken- oder Fenster-Trigger eingestellt werden.
Andere Einstellungen (inkl. Modes der Trigger-Option) könnten allenfalls direkt im Control-Panel gesetzt werden.

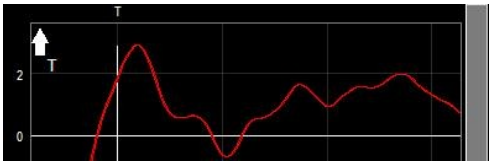


Die Pegel werden direkt im Kurven-Fenster eingestellt. Für jeden aktivierten Kanal, dessen Trigger-Mode nicht "Off" ist, erscheint am linken und rechten Fensterrand ein Dreieck-Symbol. Diese können mit der Maus gepackt und in der Höhe verschoben werden. Bei den Symbolen ist auch ersichtlich, ob auf die positive oder negative Flanke getriggert werden soll.

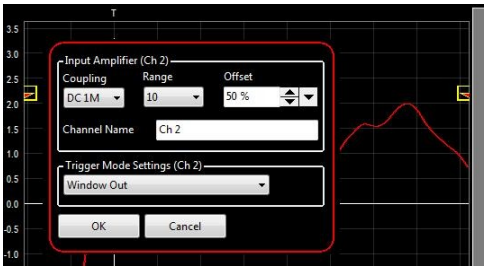


Fenster-Trigger werden durch zwei (vier) Halb-Dreiecke angezeigt. Diese können einzeln gepackt und verschoben werden.

Falls sich Pegel ausserhalb des Aufnahme-Bereichs befinden, wird ein Warndreieck angezeigt. D.h. es kann durch diesen Kanal kein Trigger ausgelöst werden.

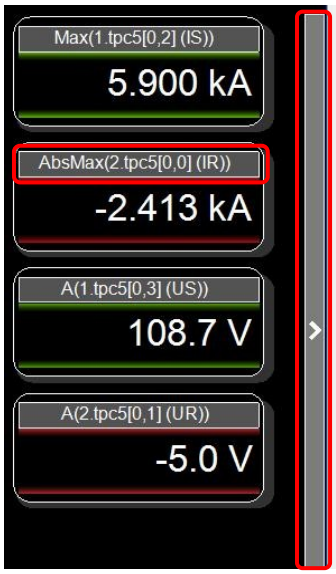


Falls sich die Pegel oben oder unten über dem Bildrand befinden (infolge Y-Zoom), wird dies durch weisse Pfeile angezeigt. Auch diese können mit der Maus gepackt und ins Fenster gezogen werden.



Wenn bei den Symbolen einfach mit der linken Maus-Taste geklickt (statt gepackt) wird, öffnet sich die Maske für die Kanal-Einstellungen (gleich, wie bei den Kanal-Namen in den Kanal-Bedienflächen).

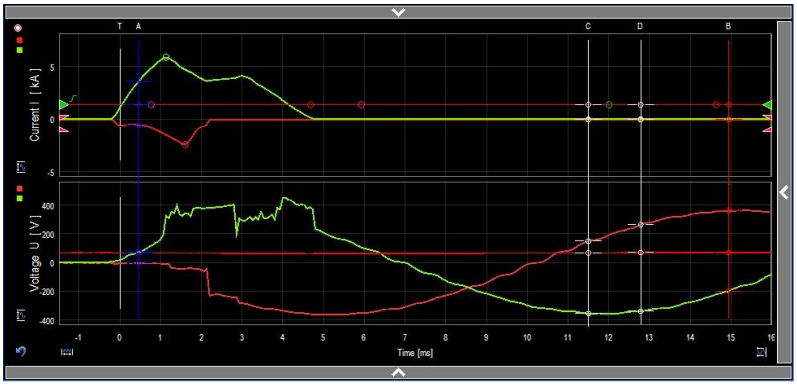
3.5 Digital-Anzeigen



Auf der rechten Seite können mehrere Anzeige-Panels dargestellt werden.
Diese lassen sich durch Klicken auf den vertikalen Balken rechts ein- oder ausblenden.
Die digitalen Wertanzeigen werden mit einer Skalar-Berechnung ermittelt. Grundsätzlich sind alle Berechnungen wie in den Skalar-Tabellen B möglich. Die Wertanzeigen sind überschrieben mit der Abkürzung des Resultats sowie dem Kanal-Namen. Wenn die Berechnung auf eine Kurve aus einer Datei angewendet wird, ist auch deren Namen eingeblendet. Mit einem Linksklick über einer solchen Überschrift kann die Maske zur Auswahl einer Skalar-Berechnung geöffnet werden.

3.6 Maximale Kurven-Darstellung

Durch einen Klick mit der linken Maus-Taste über dem oberen oder unteren Balken können die Bedienknöpfe bzw. die Kanal-Einstellungs-Flächen unterdrückt werden.

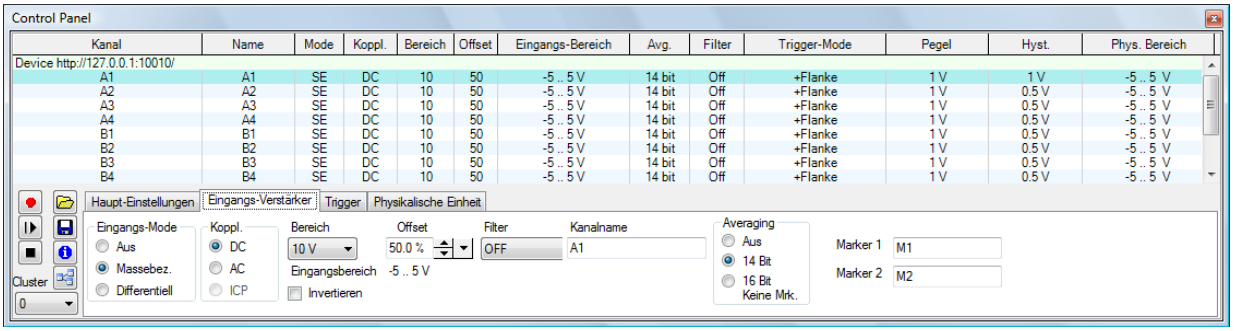


Dadurch wird der frei werdende Bereich auch der Kurven-Anzeige zur Verfügung gestellt.

© Elsys AG

4 Control Panel

Das Control Panel wird verwendet um **Aufzeichnungs-Parameter** wie Abtastrate, Eingangsverstärkung etc. einzustellen. Das Fenster ist in **zwei Bereiche** aufgeteilt:


Control Panel
Kanal
Name
Mode
Koppl.
Bereich
Offset
Eingangs-Bereich
Avg.
Filter
Trigger-Mode
Pegel
Hyst.
Phys. Bereich
Device http://127.0.0.1:10010/
A1 A1 SE DC 10 50 -5 .. 5 V 14 bit Off +Flanke 1 V 1 V -5 .. 5 V
A2 A2 SE DC 10 50 -5 .. 5 V 14 bit Off +Flanke 1 V 0.5 V -5 .. 5 V
A3 A3 SE DC 10 50 -5 .. 5 V 14 bit Off +Flanke 1 V 0.5 V -5 .. 5 V
A4 A4 SE DC 10 50 -5 .. 5 V 14 bit Off +Flanke 1 V 0.5 V -5 .. 5 V
B1 B1 SE DC 10 50 -5 .. 5 V 14 bit Off +Flanke 1 V 0.5 V -5 .. 5 V
B2 B2 SE DC 10 50 -5 .. 5 V 14 bit Off +Flanke 1 V 0.5 V -5 .. 5 V
B3 B3 SE DC 10 50 -5 .. 5 V 14 bit Off +Flanke 1 V 0.5 V -5 .. 5 V
B4 B4 SE DC 10 50 -5 .. 5 V 14 bit Off +Flanke 1 V 0.5 V -5 .. 5 V
Haupt-Einstellungen
Eingangs-Verstärker
Trigger
Physikalische Einheit
Eingangs-Mode
Aus
Massebez.
Differentiell
Koppl.
DC
AC
ICP
Bereich
10 V
Eingangsbereich -5 .. 5 V
Invertieren
Offset
50.0 %
Filter
OFF
Kanalname
A1
Averaging
Aus
14 Bit
16 Bit
Keine Mrk.
Marker 1
M1
Marker 2
M2
Cluster
0

Der **obere Bereich** enthält eine Liste mit allen vorhandenen Kanälen und deren Einstellungen. Die Kanäle werden beim Aufstarten automatisch erkannt. Man kann einen oder mehrere Kanäle auswählen um sie zu modifizieren. Falls mit gehaltener Maustaste über die Kanäle gefahren wird, werden diese markiert.
Die Signal-Kurven können von den **ersten beiden Kolonnen** in der Kanal-Liste in die Kurvenanzeige gezogen werden (**Drag & Drop**). Daher funktioniert das Markieren mit gehaltener Maustaste nicht für diese ersten beiden Kolonnen.

Die Liste im oberen Bereich des Control-Panels kann zu Dokumentations-Zwecken in eine Text-Datei gespeichert werden. Dazu sind die Kanal-Linien die gespeichert werden sollen zu markieren. Danach kann via <Strg>-S der Speichervorgang eingeleitet werden.

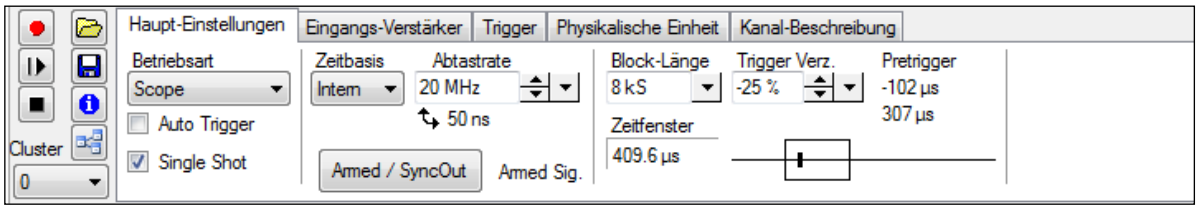
Im **unteren Bereich** des Control Panels werden die jeweiligen Einstellungen vorgenommen. Die verschieden Parameter sind in Registern nach Parameter-Klassen unterteilt.

4.1 Haupt-Einstellungen

Mit dem TranAX Datenerfassungs-System lassen sich sehr schnelle Signale (Transienten), aber auch lang andauernde, vereinzelt oder periodisch auftretende Signale digitalisiert erfassen und abspeichern.

Die Hardware-Grundausführung besteht aus einem **Datenerfassungs-Modul mit 4 oder 8 Kanälen**. Je nach Platz im Rechner können zusätzliche Erweiterungsmodule eingebaut werden. Konzeptionell ist ein Ausbau von bis zu **64 Kanälen** in einem Instrument möglich.

Die Signale können in mehreren Kanälen parallel aufgezeichnet werden, wobei jeder Kanal seinen eigenen Speicherteil besitzt, unabhängig vom Arbeitsspeicher des Computers. Die Triggerbedingung, d.h. die Voraussetzungen für das Auslösen bzw. Abschliessen einer Datenaufnahme, kann für verschiedene Kanäle individuell vorgegeben werden.
Die Datenerfassungs-Hardware, TPCX, TPCE oder TraNET, kennt grundsätzlich drei verschiedene Aufnahme-Betriebsarten.



Die Registerkarte ***Haupt-Einstellungen*** enthält folgende einstellbaren Parameter:

- Betriebsart: Die Datenerfassungs-Hardware, TPCX, TPCE oder TraNET, kennt grundsätzlich die folgenden Aufnahme-Betriebsarten.
  - Scope (mit Autotrigger und/oder Single Shot)
  - Multiblock
  - Kontinuierlich
  - ECR (Event Controlled Recording)
- Zeitbasis: Intern oder Extern
- Die Abtastrate (interne Zeitbasis) oder erwartete Taktfrequenz (externe Zeitbasis, wird für gewisse mathematische zeitrelevante Funktionen verwendet)
- Aufzeichnungslänge (Blocklänge mit Pre- und Post-Trigger)
- Triggerverzögerung (legt Pre- und Post-Trigger Verhältnis fest)

Mit dem Knopf ***"Armed / SyncOut"*** kann der entsprechende Ausgang am Digital-Stecker als Armed-Signal oder als Ausgang von Takt-Pulsen, mit einstellbarer Frequenz, geschaltet werden. Diese Frequenz ist unabhängig von der eingestellten Zeitbasis. Sie dient zur Synchronisation von externen Geräten (z.B. Highspeed-Video-Kameras).
Bei älteren Geräten ist dieser Knopf nicht vorhanden. Die Hardware dieser Geräte müsste im Lieferwerk nachgerüstet werden.

Die **Abtastrate** kann entweder als **Abtastfrequenz** oder als **Zeitintervall** angezeigt und eingegeben werden. Das Eingabefeld reagiert auf einen beigefügten Dezimalvorsatz ("m" für Milli, "k" für Kilo, "M" für Mega, "u" für Micro) und eine Einheit ("H" oder "Hz" für Hertz, "s" für Sekunden). Beispiele:

Eingabe	Interpretiert als
10**k**	10 **kHz**
2.5**M**	2.5 **MHz**
2.5**m**	2.5 **ms**

Eingabe	Interpretiert als
15**u**	15 **µs**
500**Hz**	500 **Hz**
3**s**	3 **sec**

Die Aufnahmezeit (Zeitfenster) wird von der Blocklänge und der Abtastrate bestimmt.

4.2 Icons

Die Schaltflächen links im Control Panel haben folgende Funktionen:

	Aufzeichnung starten (F6). Alternativ dazu kann die Datenerfassung gestartet werden mit einem externen TTL-Signal, das mit der Schaltfläche freigeschalten wird.
	Manueller Triggermodus (F7)
	Aufzeichnung manuell stoppen (F8)
	Einstellungen laden
	Einstellungen speichern
	Hardware Konfiguration anzeigen
	Cluster Konfiguration anzeigen

4.3 Control Panel Register

Um Einstellungen zu verändern, verwenden Sie die Dropdown Menüs und Optionen-Listen. Textfelder können entweder manuell ausgefüllt werden oder mittels folgender Schaltflächen:

	Wert erhöhen oder verringern
	Werte aus der Liste auswählen
	Wechselt die Einheit der Parameter (z.B. Frequenz/Abtastintervall)

Einige Hardware-Einstellungen werden in kleinen **Illustrationen** bei den Werten dargestellt. Sie können dann die Einstellungen auch **mittels den Markierungen ändern** wie z.B. die Triggerverzögerung.
Alle aktuellen Einstellungen können im Menü ***"Datei" / "Aufnahmeeinstellungen speichern.."*** oder unmittelbar über den Knopf auf der linken Seite des Control Panels auf die Festplatte abgelegt werden.

5 Referenzen

Ab TranAX Version 3.4.1.1200 ist es möglich, **anstelle von Kurven aus Dateien mit sog. Referenzen zu arbeiten**. Es lassen sich auch direkt Hardware-Kanäle den Referenzen zuordnen.

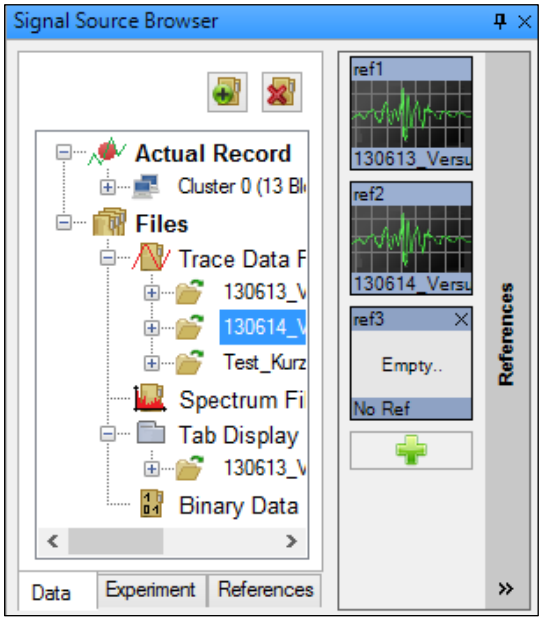
Mit den Referenzen hat man die Möglichkeit einzelne Tpc5-Dateien, welche man beispielsweise als Messreihe mit einer Autosequenz erzeugt hat, nacheinander zu analysieren. Eine Referenz kann man sich als Platzhalter vorstellen, welcher in einer Kurvenanzeige, in den Skalar-Tabellen oder im Formeleditor verwendet werden kann. Beim Zuordnen von Kurven aus einer andern Datei (oder direkt aus Hardware-Kanälen) werden nur die Daten ausgetauscht, die Eigenschaften (Farbe, Strichdicke, Zoombereich, Labels usw.) bleiben jedoch erhalten.

Dies bietet den **Vorteil**, dass das **Ausarbeiten** und **Vorbereiten** eines **Layouts**, d. h. Kurvenanzeigen, Skalar-Tabellen, platzierte Texte usw. **nur einmal mit den Referenzkurven erstellt werden muss**. Anschliessen können andere Messkurven via Drag & Drop zugewiesen werden, ohne diese im Layout oder in einer Formel jedes Mal ersetzen zu müssen.

Es werden keine Kopien der Dateien erstellt. Die Referenz verhält sich wie ein Zeiger, welcher auf die gewählte Signal-Quelle (Datei oder Hardware-Kanäle) verweist.
Wird eine solche **Quelldatei** auf dem Rechner **entfernt** oder **verschoben**, so verschwindet auch die Referenz, d. h., sie wird leer.

5.1 Mögliche Quellen für Referenzen

Die Handhabung der Referenzen geschieht im Signalquellen-Browser ( ).



Im Signalquellen Browser findet man gleich neben dem Datei-Baum die Referenz-Felder vor. Diese können mit dem vertikalen grauen Balken zur Ansicht gebracht werden.
Durch Klicken auf das Symbol können weitere (vorerst leere) Felder hinzugefügt werden.
Es lässt sich nur das jeweils unterste Feld entfernen. Man klicke dazu auf das kleine "x" oben rechts im Feld.
Um nun eine Referenz zu belegen, zieht man das gewünschte Objekt (eine gesamte Datei oder auch nur einzelne Kurven) via **Drag & Drop** auf ein Referenz-Feld.
Normalerweise wird danach dieses Referenz-Feld via **Drag & Drop** in eine Kurvenanzeige gezogen.

© Elsys AG

Folgende Objekte sind gültig:

- **Gespeicherte Dateien**, dabei werden die folgenden Formate unterstützt:
  - ***.Tpc5**: TranAX 3 Standard Format für gemessene Kurven im Zeitbereich
  - ***.TPS5**: TranAX 3 Standard Datenformat für Spektrum und FFT Kurven
  - ***.TDP**: gespeicherte Tab-Pages in TranAX 3
  - ***.BDF**: binäres Rohdatenformat von Schreiber- und ECR-Mode
- **Hardware Kanäle**, alle Kanäle aus dem Control Panel bzw. aus einem aktuellen Cluster, dabei können auch mehrere selektiert und dann via Drag & Drop aus den ersten beiden Kolonnen im Control Panel auf ein Referenz-Feld gelegt werden.
- **Gemittelte YT- und FFT-Kurven** (@- bzw. %-Kurven), hier können ebenfalls mehrere Kanäle selektiert und dann auf ein Referenz-Feld gezogen werden.

Die belegten Referenz-Felder können dann ebenfalls via Drag & Drop in eine Kurvenanzeige gezogen werden. Dies betrifft immer all die Kurven, welche auf die entsprechende Referenz gelegt wurden. Wenn also z. B. nur eine Kurve behandelt werden soll, muss auch nur diese eine Kurve auf das Referenz-Feld gezogen werden.

5.2 Referenzen im Formeleditor

Im Formeleditor kann ebenfalls mit den Referenzen gearbeitet werden.
Mit dem File-Befehl kann auf eine Kurve in einer Referenz zugegriffen werden.

```
trace = File(ref1,index) [.blkNo] ['markerNo]
```

Der Index bezeichnet die Kurve in der Datei (0…)). Anstelle des kompletten Datei-Namens in Anführungs-Zeichen muss nur die Bezeichnung ***ref*** mit der entsprechenden Nummer angegeben werden.

Falls auf die erste (nullte) Kurve in einer Referenz zugegriffen werden muss, kann die Formel auch vereinfacht werden:

```
trace = ref2 [.blkNo] ['markerNo]
```

Diese Anweisung entspricht trace**=File(ref2,0)**

Wie üblich können fakultativ auch Block- und/oder Marker-Nummern angegeben werden.

Folgende Funktionen sind ebenfalls zulässig:

```
val = FileIndexExist (ref1, 2) ; val = True oder False
nTrc1 = NTracesInFile (ref1); ist gleichbedeutend wie Length (ref1)
nTrc2 = Length (ref2)
```

Ref wird im Formeleditor als Schlüsselwort verwendet.

5.3 Referenzen in der Autosequenz

Referenzen können auch in den Autosequenzen verwendet werden. Beim Befehl **Save** kann anstelle des Dateinamens die entsprechende Referenz (**ohne Namenserweiterung** z. B. TPC5 usw.) angegeben werden:

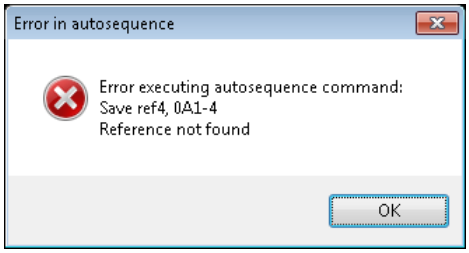
```
Save ref1, 0A1-4
```

Mit dieser Methode ist es möglich in einem Autosequenz-Loop einerseits gemessene Kurven zu archivieren (auf Dateien mit fortlaufender Nummerierung) und gleichzeitig die aktuellen Messdaten für die Anzeige oder die Weiterberechnung im Formeleditor zu verwenden.

Beispiel einer solchen Autosequenz:

```
Repeat 10
Start Recording
Wait on EOR
Save xy_#.tpc5, 0A1-4   ; Erzeugen einer Messreihe
Save ref4, 0A1-4        ; Erzeugen (Überschreiben) einer Datei
                        ; "ref4.tpc5" und Zuweisen zu der
                        ; entsprechenden Referenz
Calculate
Wait for Calculations
Next
```

Der Ausdruck "ref" dient als Schlüsselwort und wird entsprechend geprüft!
Sollte das Referenz-Feld mit der entsprechenden Nummer (z. B. "ref4") im Signal Source Browser nicht vorhanden sein, erscheint eine Fehlermeldung:



Durch Klicken auf das Symbol im Signal Source Browser können weitere Referenz-Felder hinzugefügt werden.

In der **Autosequenz** wird die **Kurve ebenfalls physikalisch als Datei** (z. B. "ref1.tpc5") im **Verzeichnis "data" des aktuellen Experiments abgelegt** und der entsprechenden Referenz zugewiesen.

6 Aufnahme-Betriebsarten

6.1 Scope

Der Scope-Mode ist der Grundeinstellungs-Mode. Es werden keine Daten auf die Festplatte gespeichert. Die aktuelle Aufzeichnung ist nur im Speicher der TPCX/TPCE-Module vorhanden. Nachdem die Aufzeichnung gestartet und abgeschlossen wurde, wird der aktuelle Block angezeigt.
In dieser Registerkarte kann man die Zeitbasis, Abtastrate, Blocklänge und Triggerverzögerung eingeben. Die Illustration unterhalb der Triggerverzögerungs-Eingabe zeigt den Trigger Punkt grafisch in Relation zu der gesamten Aufzeichnungslänge. Die Triggerverzögerung lässt sich auch grafisch mit der Maus einstellen.

☑ Auto Trigger	Falls kein Trigger-Ereignis eintritt, triggert TranAX automatisch.
☑ Single Shot	Nur ein einziges Ereignis wird angezeigt und die Aufzeichnung beendet.

Falls mehr als nur ein Cluster konfiguriert ist, ist die Checkbox **Single Shot** immer aktiviert (Auto-Start Aufnahme ist nicht möglich).
Alternativ kann eine Auto Sequenz mit dem Kommando "Start Recording" in einer Schlaufe verwendet werden.

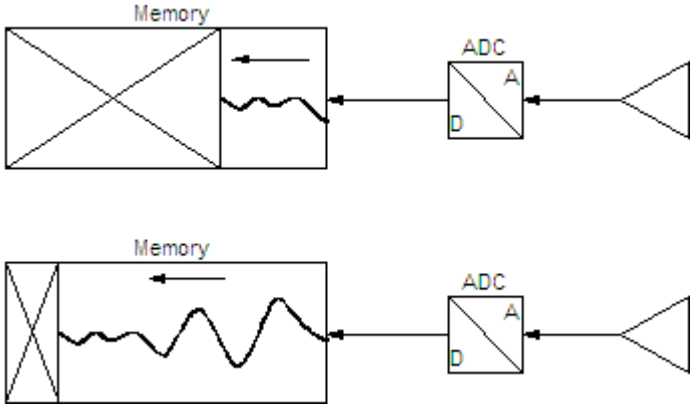
6.2 Multiblock

Der Multiblock-Mode speichert die aufgezeichneten **Signale fortlaufend** in Segmente des Kanal-Speichers. **Jedes Trigger-Ereignis wird in einem Block gespeichert**. In dieser Registerkarte können auch die **Zeitbasis**, die **Abtastrate**, die **Blocklänge** und die Triggerverzögerung eingestellt werden. Zusätzlich zum Scope-Mode können hier die Anzahl Blöcke bzw. Anzahl der Aufzeichnungen festgelegt werden. Die maximale Anzahl an Blöcken hängt von der Blocklänge und vom gesamten Modul-Speicher ab.

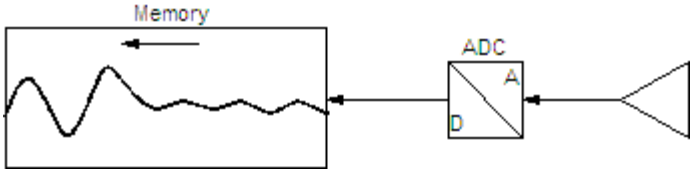
Der Multiblock-Mode ist speziell geeignet für das Erfassen von Bursts mit einer **hohen Repetitionsrate** und **minimaler Totzeit** zwischen den Bursts. Falls die Anwendung eine totzeitlose Erfassung von Bursts verlangt, so ist der ECR-Mode der ideale Datenerfassungs-Mode.

Im Multiblock-Mode übernehmen die TPCX/TPCE-Module oder das TraNET-Messgerät vollständig die Kontrolle über den Ablauf und die Speicherung einer Messung. Die Messung hat eine vorher **festgelegte Speicherlänge (Blockgrösse)**, die in den auf den Modulen verfügbaren Arbeitsspeicher passen muss. Der Rechner ist an der Messung nicht beteiligt, er gibt nur das Start-Kommando und wartet bis die Messung beendet wurde. Danach liest TranAX die Daten aus. Im Folgenden wird das Prinzip einer Messung im Block-Mode beschrieben.

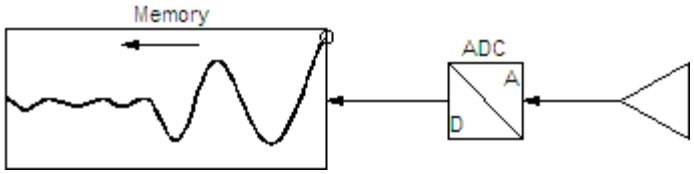
Die TPCX/TPCE-Module beginnen sofort nach dem **Start der Messung** damit, ihre Speicher mit **Werten von den ADCs** zu füllen, bis die eingestellte **Blockgrösse erreicht** wurde.



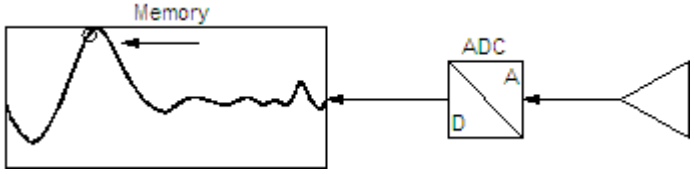
Von jetzt an werden die **ältesten Werte** fortlaufend durch die neuen **überschrieben**.



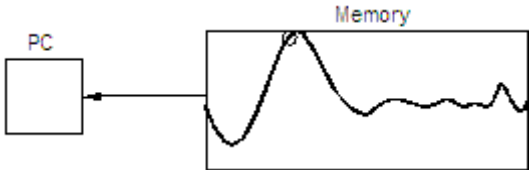
Durch dieses Verfahren liegt immer das letzte Stück des Signals aus der Vergangenheit bis zur **Gegenwart** im Speicher. Tritt ein **Trigger-Ereignis** ein, kann die Aufnahme angehalten werden, und man hat den Signalverlauf **vor dem Triggerpunkt**.



Falls man am weiteren Signalverlauf **nach dem Triggerpunkt** interessiert ist, kann die TPCX/TPCE-Hardware noch eine Weile weiteraufzeichnen, bevor die Aufnahme beendet wird. Siehe auch Pre- und Post-Trigger (Triggerverzögerung).



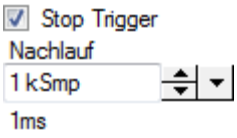
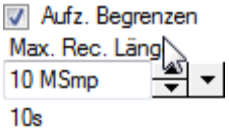
Nachdem die Aufnahme beendet wurde, werden die aufgenommenen Daten aus dem Modulspeicher an den PC übertragen.



6.3 Kontinuierlich

Im kontinuierlichen Mode arbeitet der Transientenrekorder als **Disk-Rekorder**. Es können keine Blocklängen oder Pre/Post Trigger-Einstellungen gemacht werden.

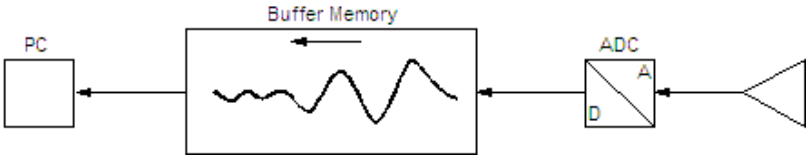
Die Aufzeichnung wird mittels der Start-Schaltfläche gestartet. Um die Aufzeichnung zu beenden gibt es drei Möglichkeiten:

	Die **Aufzeichnung wird beendet**, falls ein Trigger-Ereignis auftritt. Zusätzlich kann eine Nachlaufzeit festgelegt werden. Die Aufzeichnung wird nach einem Trigger-Ereignis noch um eine voreingestellte Dauer fortgesetzt. Die Trigger-Einstellungen werden in der Trigger Registerkarte eingegeben.
	Die Aufzeichnung wird nach einer bestimmten **Zeitdauer automatisch beendet**, falls dies nicht durch einen allfälligen Stop-Trigger vorher erfolgt ist. Die Aufzeichnungsdauer kann auch vom verbleibenden Platz auf der Festplatte abhängen.
	Die Aufzeichnung kann manuell gestoppt werden.

Falls beide Optionen aktiviert sind, wird die Aufzeichnung beendet sobald eine der Bedingungen eingetreten ist.

Bitte beachten: Falls ein anderer Cluster einen Trigger auslöst, wird der Stop Trigger auch im kontinuierlichen Mode ausgelöst!

Im kontinuierlichen Mode werden alle **aufgezeichneten Werte** sofort an den Computer übertragen. Dieser speichert sie auf die **Festplatte**. Die Aufzeichnung kann vom Computer aus oder durch die TPCX/TPCE-Hardware beendet werden, wenn diese ein Trigger-Ereignis erkennt. Die **Modul-Speicher** dienen in diesem Mode als **Buffer** für die gemessenen Werte.



Die **Aufnahmelänge** ist praktisch nur durch die **Festplatte begrenzt**. Durch die begrenzte Daten-Transferrate zwischen den Modulen und dem Computer sind jedoch der Abtastrate Grenzen gesetzt. Je nach Rechner kann die maximale **Summenabtastrate** bis zu einigen Dutzend Mega Samples pro Sekunde betragen. Diese **hohe Aufzeichnungsrate** wird erreicht, wenn sichergestellt ist, dass der Computer nicht durch fremde Programme oder Prozesse belastet wird. Bei reduzierter Abtastrate ist jedoch auch bei sporadischer Überlastung der CPU eine lückenlose Aufzeichnung sichergestellt, weil die Messwerte zuerst im Modul-Speicher gepuffert werden. Dieser **Aufzeichnungsmode** ist hauptsächlich für **lang andauernde, lückenlose Messungen** interessant.

Falls die Festplatte voll wird, wird die Aufzeichnung automatisch abgebrochen.

6.4 ECR Mode

Der ECR Mode ist optional erhältlich.

Der ECR Aufzeichnungs-Mode ermöglicht die gezielte Erfassung von **zyklisch** oder **sporadisch** auftretenden **Ereignissen**. D.h. die Registrierung von Messdaten erfolgt nur dann, wenn gewisse **Signal-Bedingungen** (Trigger, Zeitfenster, Wiederholungen usw.) erfüllt sind. Dadurch werden nur benötigte Signaldaten gespeichert.
Im ECR-Mode kann garantiert werden, dass keine Totzeiten auftreten und somit keine Ereignisse oder Information verloren gehen. Dies gilt auch, wenn sehr viele Kanäle mit maximaler Abtastrate über eine beliebig lange Zeit überwacht werden müssen. Da jeder Kanal seinen eigenen Signalpuffer (bis zu 64 MSamples) besitzt, darf lediglich die mittlere Anzahl der Ereignisse pro Sekunde einen gewissen Wert nicht überschreiten. Dieser Wert hängt von der einstellbaren Blocklänge pro Ereignis ab und ist zudem durch den max. möglichen Datendurchsatz direkt auf die Festplatte bestimmt (**je nach CPU/Disk-System ca. 20 MSamples pro Sekunde**). Die Triggerbedingungen können für jeden Kanal individuell eingestellt werden, wobei sich auch komplexe Kriterien des Signalverlaufs definieren lassen (Advanced Trigger Option: Pulsbreite/Höhe, Periode, Steilheit, Fenster-Ein/Austritt usw.).

Im Gegensatz zum Block-Mode sind **Null-Totzeiten garantiert**. Bitte beachten Sie, dass im Block-Mode ein Trigger-Ereignis nicht aufgezeichnet wird, falls das Ereignis genau am Ende des Blocks auftritt. Der Bereich der Überlappung hängt von der Ereignis-Wiederholrate ab und kann in gewissen Grenzen mit Holdoff oder Retrigger gesteuert werden.

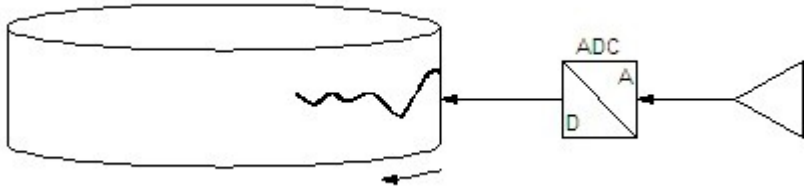
In der ECR-Betriebsart werden die anliegenden **Signalkurven laufend** in den als **Ringbuffer** geschalteten Modulspeicher geschrieben. Danach markiert jedes **Trigger-Ereignis** den **Adressbereich**, woraus die Signaldaten aus dem Modul-Speicher ausgelesen und in eine Datei geschrieben werden. Diese Aufzeichnungsart entspricht weitgehend dem bekannten Multiblock-Mode. Die **aufgezeichneten Zeitabschnitte** können sich hier jedoch **überlappen**. Nur dadurch ist eine Datenerfassung ohne Totzeit gewährleistet (vor allem bei Aufnahmen mit Pre-Trigger).

Falls man im ECR Mode ist, wird eine ECR Registerkarte angefügt. Die Blocklänge wird explizit via Pre- und Post-Trigger-Einstellungen ermittelt. Wie im Multi Block Mode kann die maximale Anzahl an Blöcken festgelegt werden, die aufgezeichnet werden sollen. Weiterhin wird ein Retrigger (RT) Marker oder ein Holdoff (HO) Marker unterhalb der Einstellungen angezeigt, je nach gemachten Einstellungen in der ECR Registerkarte.
Es gibt zwei verschiedene ECR Modi: Einzelkanal und Multikanal. Beide Modi unterstützen die Option Dual Mode.

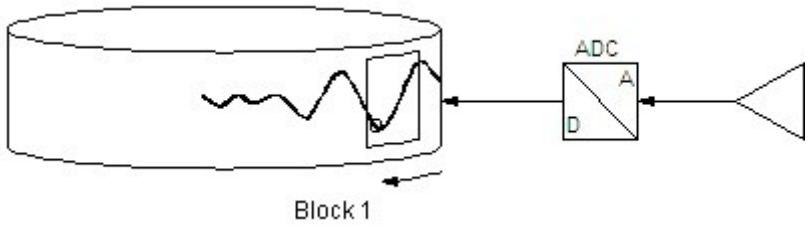
Falls die Trigger-Bedingungen etwas unkritisch gesetzt sind, kann es sein, dass die CPU von schnellen periodischen Signalen überbelastet wird. Das erweckt den Eindruck, als ob die CPU blockiert sei.

6.4.1 Prinzipieller Ablauf

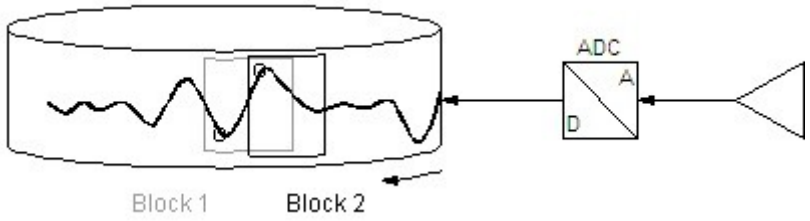
Der ECR-Mode läuft wie folgt: Das digitalisierte Signal wird auf dem internen Kartenspeicher (als **Ringbuffer** programmiert) gespeichert.



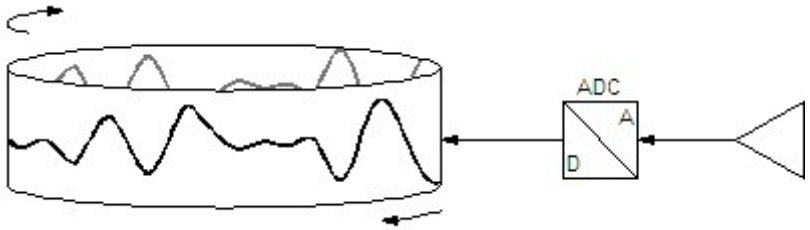
Sobald ein **Trigger-Ereignis** eintritt, werden die Block Samples vom Ringbuffer auf die **Festplatte gespeichert**.



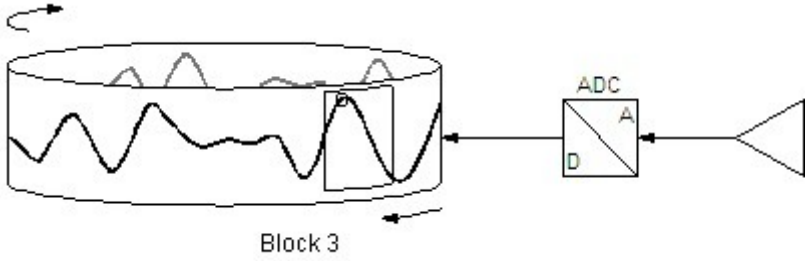
Falls ein **weiteres Trigger-Ereignis** während dem laufenden Block eintritt, wird ein überlappender Block aufgezeichnet.



Die **ältesten Aufzeichnungen** werden mit den neu **ankommenden Daten überschrieben**, sobald der Ringbuffer voll ist.
Normalerweise sind diese Daten **vorher** ausgelesen und auf die **Festplatte** transferiert worden. Falls zu viele Ereignisse gleichzeitig auftreten, kann es vorkommen, dass der **Ringbuffer überläuft**. TranAX zeigt dann eine entsprechende Statusmeldung an.



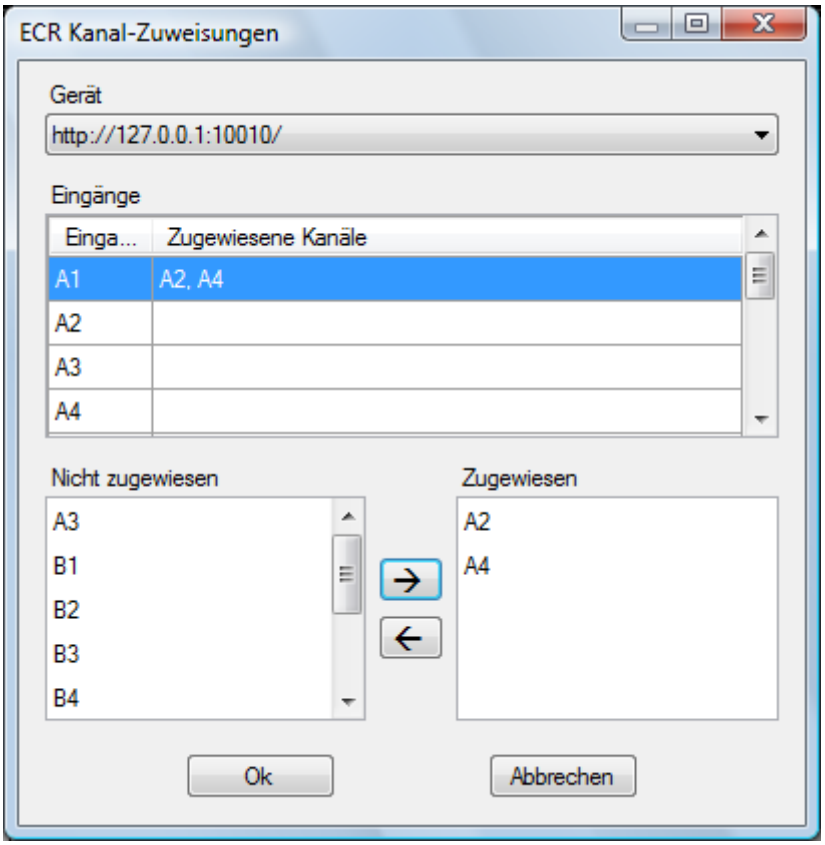
Nachdem die Anzahl eingestellter Blocks (in diesem Beispiel 3) erreicht wurde, wird die Aufzeichnung beendet.



6.4.2 ECR Single Channel Mode

Bei ECR Single Channel werden die Daten kanalbezogen registriert.

In diesem Mode kann **jeder Kanal als Einzel-Trigger-Quelle** gewählt werden. Es werden nur die Daten aus den entsprechenden Kanälen unabhängig voneinander registriert. Jedoch kann jedem **einzelnen Kanal assoziierte Kanäle zugeordnet werden**. Die Signale dieser assoziierten Kanäle werden dann im gleichen Zeitausschnitt zu den Signalen des triggernden Kanals abgelegt.
Durch Drücken auf **"ECR Assoziierung"** können Kanäle zugewiesen werden. Dabei erscheint folgendes Fenster:



Im oberen Teil in der Spalte "Eingang" kann der gewünschte Kanal gewählt werden. Danach können im unteren Feld mit dem rechtem Pfeil die zuzuweisenden Kanäle bestimmt werden.

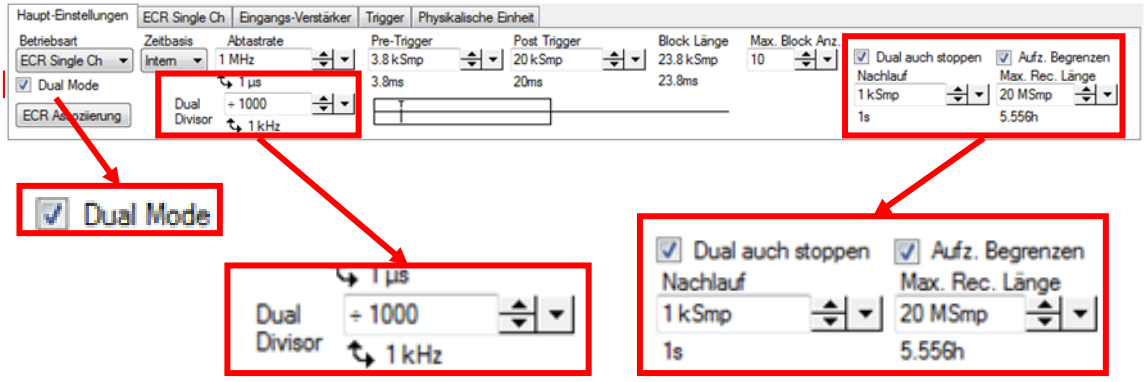
6.4.3 ECR Multi Channel Mode

Im ECR Multi Channel Mode werden die Signaldaten parallel aus allen eingeschalteten Kanälen aufgezeichnet.

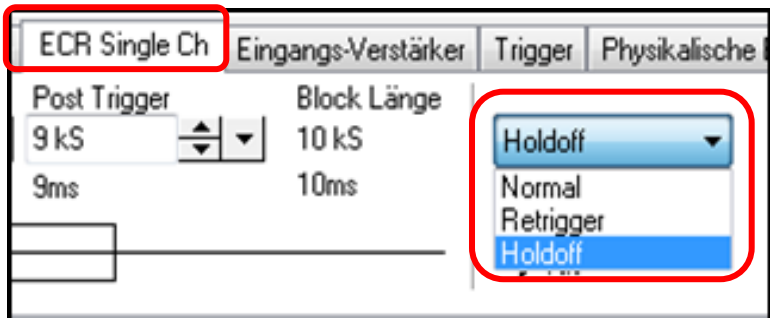
6.4.4 Dual Mode

Durch Zuschalten von **Dual Mode** wird der ECR-Mode **erweitert**. Dabei werden parallel zur Registrierung von schnell aufgezeichneten ECR Trigger-Ereignissen die **Signale von allen eingeschalteten Kanälen** kontinuierlich, **in der Regel relativ langsame Kurvenverläufe**, mit aufgenommen. Mit dem Parameter **Dual Divisor** lässt sich die Taktrate für die langsamere, kontinuierliche Aufzeichnung im Verhältnis zur schnellen Abtastrate einstellen.

Die kontinuierliche Aufzeichnung läuft bis auf die Start- und Stopp-Bedingung synchron zu den allenfalls auftretenden ECR-Ereignissen. Die langsame Aufzeichnung speichert immer die Signale aus allen eingeschalteten Kanälen in die Datei. Dies ist auch der Fall bei ECR-Single Channel. Es empfiehlt sich deshalb, nicht verwendete Kanäle auszuschalten.



6.5 ECR Trigger Optionen

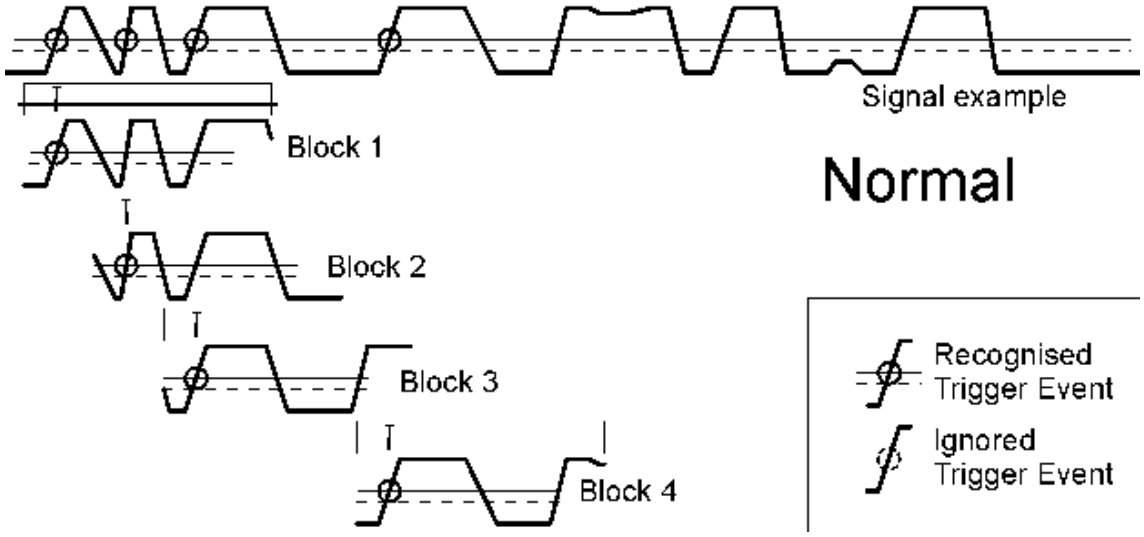


Zusätzlich zu den ECR-Einstellungen in der Registerkarte Haupt-Einstellungen, können in der Registerkarte ***"ECR Single Ch"*** beziehungsweise ***"ECR Multi Ch"*** die Optionen
Normal, ***Retrigger*** und ***Holdoff*** gewählt werden.

Pre-Trigger, *Post-Trigger* und *Anzahl Blöcke* können entweder unter *Haupt-Einstellungen* oder in der *ECR*-Registerkarte eingestellt werden.

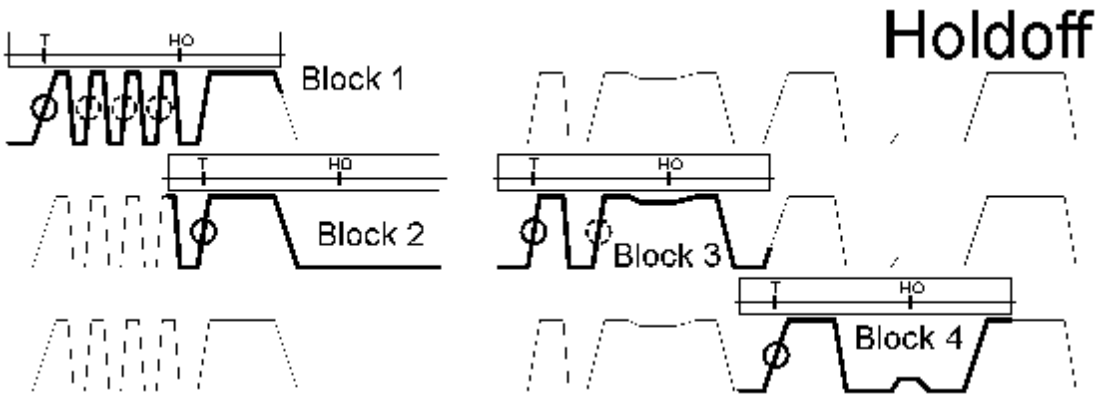
6.5.1 Normal

Falls die Option ***Normal*** gewählt wird, können **keine weiteren Einstellungen** gemacht werden. Diese Einstellung sollte möglichst vermieden werden, da sie leicht zur Überlastung des Computers führen kann.
Mit der zusätzlichen Option *Retrigger* oder *Holdoff* kann die **Überlappung der Datenblöcke optimiert** werden.



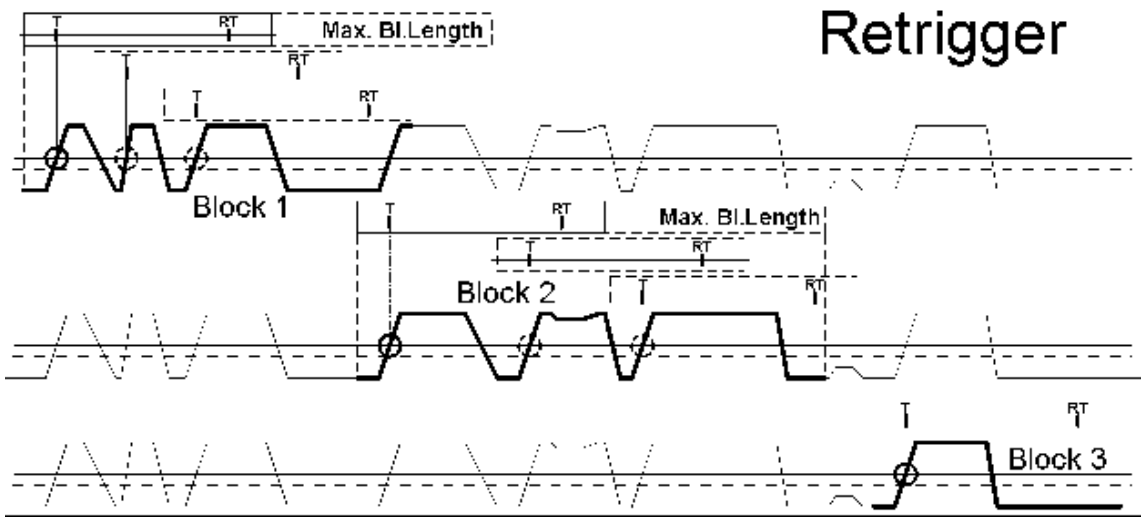
6.5.2 Holdoff

Mit der Holdoff Trigger-Option wird TranAX alle zusätzlichen **Trigger-Ereignisse bis zur *Holdoff*-Marke ignorieren**.

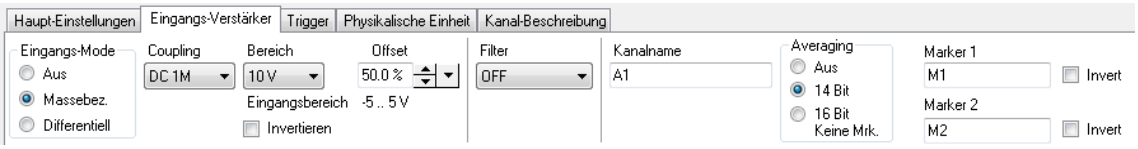


6.5.3 Retrigger

Bei Retrigger (wie auch bei Holdoff) kann die Marke entweder in **Sample-Längen** oder als **Zeitabstand** eingestellt werden. In diesem Zeitabschnitt werden **Trigger-Ereignisse ignoriert und nicht aufgezeichnet**. Falls in dieser Zeit aber ein neues Ereignis eintritt, wird die Retrigger-Marke relativ zu diesem Ereignis **neu gesetzt**. D.h. es erfolgt ein Retrigger und die Blocklänge wird vergrössert. Erst nachdem die Marke ohne Trigger-Ereignis überschritten wurde, kann ein neuer Block begonnen werden. Es kann eine maximale Posttrigger-Blocklänge eingestellt werden. TranAX triggert im Retrigger-Mode wie folgt:



7 Eingangsverstärker



Auf dieser Registerkarte stellt man folgende Kanalparameter ein:

- **Verstärker Mode**: Single-Ended (Massebezogen), Differentiell oder Ausgeschaltet
- **Kopplung**: DC, AC oder ICP (Integrated Current Power für Piezo-Sensoren) Bei den 120MS und 240MS Modulen (TPCX und TPCE) kann zusätzlich die Eingangsimpedanz auf 50Ω gestellt werden. Bei allen anderen Modulen beträgt diese 1MΩ.
- **Eingangsspannungsbereiche**: Gesamter Bereich und Offset
- Eingangs-Tiefpass-**Filter** (inkl. Anti-Aliasing-Filter-Frequenz, als Option erhältlich)
- **Invertieren** des Eingangssignales
- **Kanal-Name**
- **Averaging**: Aus, 14Bit oder 16Bit
- **Namen** von **Marker**-Signalen (optionale Digitaleingänge)
- **Invertieren** der **Marker**-Signale

Diese Werte können für **jeden Kanal individuell** eingestellt werden. Beim Programmstart erkennt die Software automatisch, welche Kanäle im System installiert sind. Diese werden in der Tabelle aufgelistet.

7.1 Mittelung (Averaging)

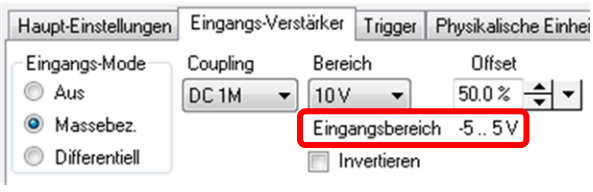
Der ADC läuft immer mit maximaler Abtastrate. Falls die gewählte Abtastrate unterhalb der maximalen Rate ist, werden alle Samples reduziert und gemittelt. **Das Signalrauschen wird entsprechend verkleinert**.

Der Parameter **Averaging** wirkt immer gemeinsam auf **alle Kanäle** innerhalb eines Modules.

Für Anwendungen welche eine Mittelung nicht erlauben (z.B. **Unterabtastung**), muss die **Mittelung abgeschaltet** werden.

7.2 Verstärker-Einstellungen

Der **Eingangsspannungsbereich** ergibt sich aus den zwei Werten **Bereich** und **Offset**. Der Bereich gibt den maximal erfassbaren Signalbereich an. Der Offset dient dazu, den Nullpunkt des Bereichs zu verschieben. Die resultierende minimal und maximal erfassbare Eingangsspannung wird sowohl unter den beiden Eingabefeldern als auch in der Tabelle als Information angezeigt. Der **Eingangsverstärker** von jedem Kanal kann **invertierend** arbeiten, d.h. die Polarität der Eingangsspannung wird umgekehrt. Um das entsprechende Signal zu bezeichnen, kann jedem Kanal ein **individueller Name** zugewiesen werden.



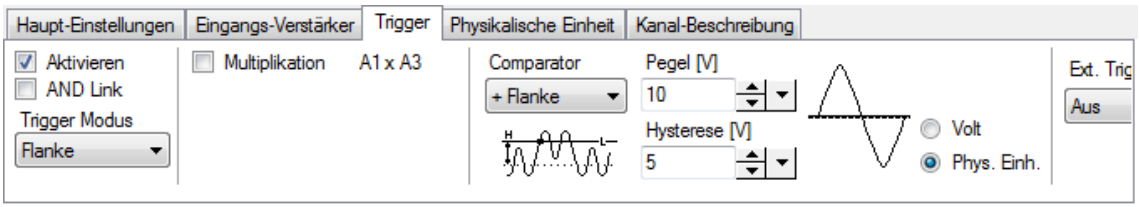
7.3 Marker (Digital-Eingänge)

Marker-Eingänge können optional installiert werden.

Jeder Datenerfassungs-Kanal kann zwei Digital-Eingänge (Marker) haben. Das sind Eingänge die ein 0/1-Signal aufweisen. Diese Signale können danach in einem **Markerfenster** dargestellt werden.
In den Eingabefeldern Marker 1 und Marker 2 können die Namen der digitalen Eingänge definiert werden. Falls "Invert" aktiviert ist, wird das Signal vor der Erfassung invertiert. Dies ist insbesondere dann von Nutzen, wenn das Signal vorher von z.B. einem Optokoppler invertiert wurde. Siehe auch Kapitel Einschränkungen / Digitale Eingänge (Marker).

	Invertierte Marker werden mit einem **"\"** (Backslash, Rückwärtsschrägstrich) am Ende gekennzeichnet. "M1\" ist der invertierte Marker 1.
	Die Marker-Einstellungen können vorgenommen, werden auch wenn keine Marker-Option installiert ist.
	Die Marker-Eingänge werden unterdrückt falls **Averaging auf 16-Bit** gesetzt ist.
	Der entsprechende analoge Kanal muss im **Control Panel eingeschalten** sein (Karteireiter "Eingangs-Verstärker"), damit die Marker aufgezeichnet werden.

8 Trigger

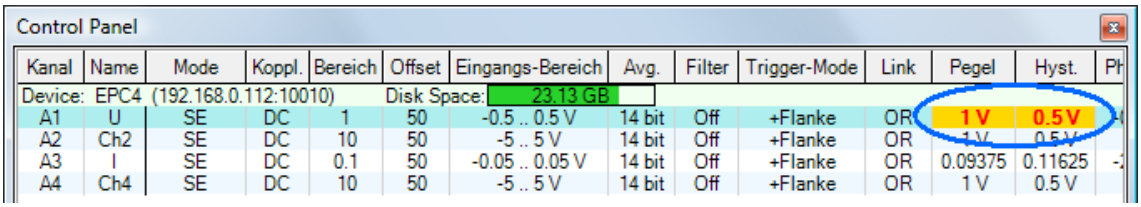


Die folgenden Parameter werden in dieser Registerkarte eingestellt:

- **Trigger Aktivieren: Der angewählte Kanal** ist nur eine aktive Triggerquelle, falls Aktivieren eingeschalten ist.
- AND Link: Die UND-Trigger-Verknüpfung wirkt nur innerhalb eines 4- oder 8-Kanal-Moduls.
- Trigger-Modi: Siehe das Kapitel Trigger-Modi für eine detaillierte Beschreibung der Trigger-Arten
- Multiplikation
- **Comparator** (Flanke)
- **Trigger-Pegel** [Volt oder Physikalische Einheit]
- **Trigger Hysteresis**

Die Trigger-Einstellungen gehören zu den sehr kritischen Parametern. Mangelhafte Einstellungen können zu ungewollten Aufnahmen führen oder die eingestellten Bedingungen sind nicht geeignet, um einen Trigger auszulösen.

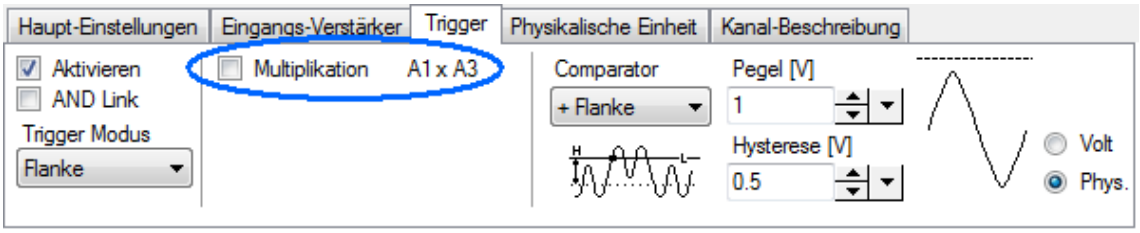
Überprüfen Sie immer zuerst genau die Trigger-Einstellungen, falls ungewollte Aufnahmen gemacht werden oder die Triggerbedingung nicht erfüllt ist. Es empfiehlt sich, alle **ungewollten Triggerquellen zu deaktivieren**.
Die Trigger-Parameter können für jede Triggerquelle separat eingestellt werden. Bei Programmstart erkennt die Software automatisch, welche triggerfähigen Kanäle im System installiert sind. Die triggerfähigen Kanäle haben in der Tabelle Einträge unter Trigger-Modus und -Pegel.
Je nach gewähltem Trigger Modus können entweder Pegel und Hysterese (+Flanke, -Flanke, ±Flanke) oder ein unterer und ein oberer Pegel (Fenster Eintritt, Fenster Austritt) in Volt oder in der physikalischen Einheit eingegeben werden. Mit der Hysterese-Einstellung kann verhindert werden, dass bei verrauschten Signalen auf die falsche Flanke getriggert wird. Die aktuellen Einstellungen werden zur Information in den Grafik-Symbolen dargestellt.


Control Panel
Kanal Name Mode Koppl. Bereich Offset Eingangs-Bereich Avg. Filter Trigger-Mode Link Pegel Hyst.
Device: EPC4 (192.168.0.112:10010) Disk Space: 23.13 GB
A1 U SE DC 1 50 -0.5 .. 0.5 V 14 bit Off +Flanke OR 1 V 0.5 V
A2 Ch2 SE DC 10 50 -5 .. 5 V 14 bit Off +Flanke OR 1 V 0.5 V
A3 I SE DC 0.1 50 -0.05 .. 0.05 V 14 bit Off +Flanke OR 0.09375 0.11625
A4 Ch4 SE DC 10 50 -5 .. 5 V 14 bit Off +Flanke OR 1 V 0.5 V

Falls die Spalten Pegel und Hysterese gelb/rot erscheinen, heisst das, dass die Einstellungen ungültig sind, z.B. ist der Eingangsbereich bis 0.5 V und der Pegel wurde bei 1 V eingestellt, d.h. dieser Pegel kann gar nie erreicht werden.

© Elsys AG

8.1 Eingangs-Multiplizierer



Die Aktivierung des Parameters **Multiplikation** bewirkt, dass die Hardware laufend die **digitalisierten Signale von zwei Kanälen** (z.B. A1 und A3 oder A2 und A4) miteinander multipliziert. Das daraus resultierende Produkt-Signal wird dann anstelle des Original-Signals von Kanal A1 bzw. A2 zum Trigger-Diskriminator geführt. Der Bereich und die Auflösung des Produkts hängt von den Einstellungen "Eingangs-Bereich", "Offset", "physikalischer Skalierungsfaktor" und "physikalische Einheit" der beiden zugehörigen Kanäle ab.
Der volle Bereich und damit die Auflösung kann nur maximal bis zur Hälfte ausgeschöpft werden. Dies ist der Fall, wenn bei beiden Kanälen der Offset auf 0% oder auf 100% gestellt ist. Je nachdem ergibt das dann eine Produkt-Kurve von 0 bis +Max oder bis -Max. Eine Produkt-Kurve von -Max bis +Max kann es nie geben.
Man erreicht nur ein Viertel der maximalen möglichen Auflösung, wenn bei beiden Kanälen der Offset auf 50% (=Bereichsmitte) gestellt ist.

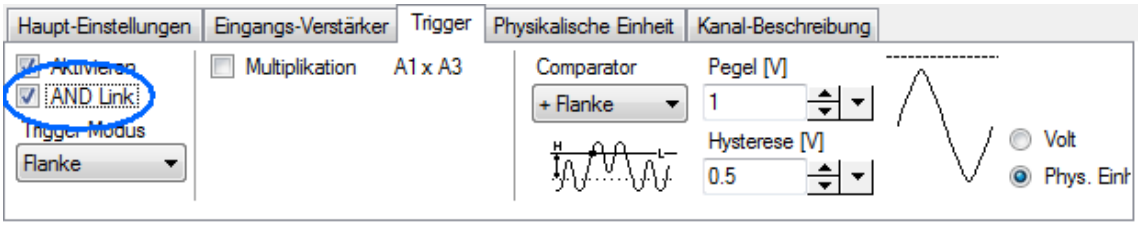
Es gilt zu beachten, dass der Hardware-Multiplizierer die Konstante bei der physikalischen Skalierung nicht berücksichtigen kann. Die Konstante sollte deshalb bei beiden Kanälen auf 0 gestellt werden.

Durch Einschalten eines weiteren Parameters **Produkt speichern** wird beim ersten der beiden Kanäle (z.B. A1) die berechnete Signalkurve anstelle des Original-Signals aufgezeichnet. Beim zweiten zugehörigen Kanal (z.B. A3) erfolgt die Aufzeichnung unverändert.

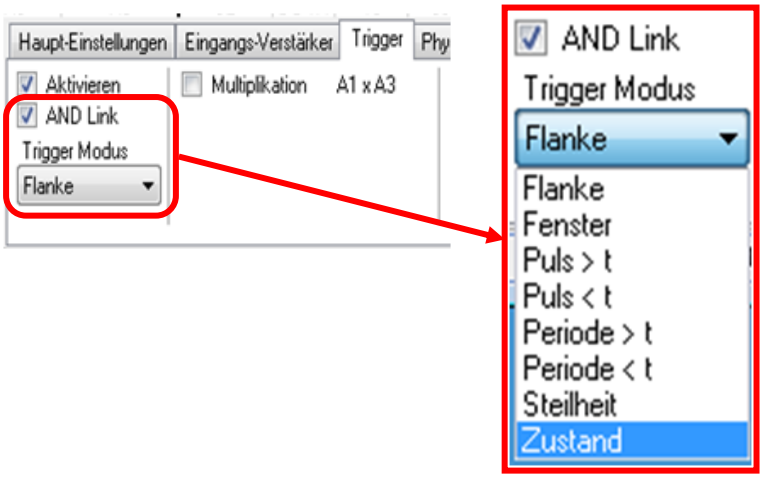
Die Amplituden-Auflösung der Produkt-Kurve beträgt auch 14-Bit bzw. 16-Bit. Falls die Original-Signale nicht den vollen Aussteuerungsbereich der Verstärker ausnützen, kann sich die Auflösung der resultierenden Kurve unakzeptierbar stark verringern. Es empfiehlt sich also für beide Signale jeweils den **Messbereich** so zu wählen, dass die **Verstärker knapp nicht übersteuert** werden.

8.2 AND Link (Logische UND Verknüpfung)

Die UND-Verknüpfung steht nur zur Verfügung, wenn die Option Advanced Trigger installiert ist.



Standardmässig sind alle aktiven Triggerquellen in ODER-Logik verknüpft. Das heisst, dass **jede aktive Triggerquelle** innerhalb eines Systems **Trigger auslösen kann**.
Falls für spezielle Triggerbedingungen die UND-Logik erwünscht ist, so können alle Kanäle eines Moduls, wie zum Beispiel A1 - A8, in diesem Fall die Eingangskanäle eines 8-Kanal TPCX/TPCE-Modules, in einer UND-Kombination konfiguriert werden. Die UND-Verknüpfung lässt sich beliebig kombinieren mit all den anderen zur Verfügung stehenden Trigger Modi.



Der Trigger Modus **Zustand** steht zur Auswahl, sobald der **AND Link aktiviert** ist.

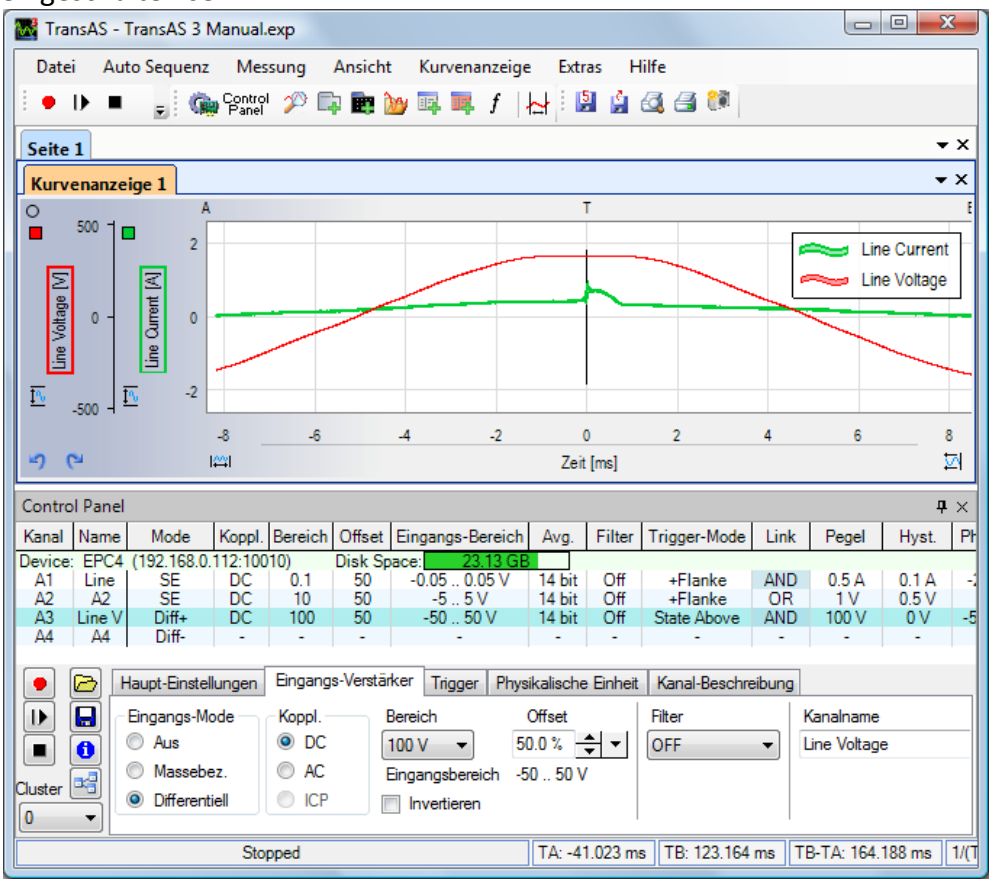
Der **Zustands-Trigger** kann als **Qualifizierer** für weitere Triggerquellen eingesetzt werden. Nur wenn der Zustand dieses Kanales zutrifft wird auf den **andern** (UND-Verknüpften) **Trigger-Quellen getriggert** sobald eine von deren Bedingungen erfüllt ist.

Falls nur bei **einem Kanal "AND Link"** aktiviert wurde, so verhält sich diese Trigger-Quelle im System genau gleich wie bei einer ODER-Verknüpfung.
Es müssen **mindestens zwei Kanäle** "AND Link" zugeschaltet haben um diese Funktion sinnvoll zu verwenden.

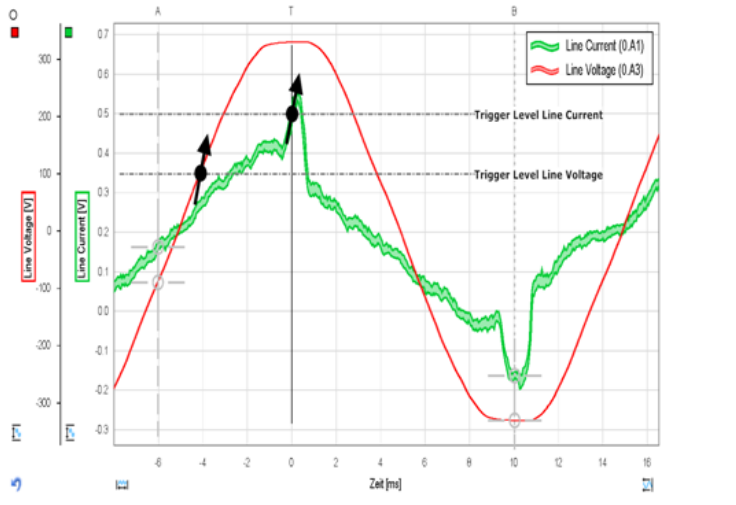
Die Standard ODER-Verknüpfung funktioniert mit allen aktiven Trigger-Kanälen, auch bei mehreren Transientenrekordern, die via Sync-Link Box synchronisiert sind. Die UND-Verknüpfung hingegen lässt sich ausschliesslich pro Kanalgruppe aufsetzen, also pro 4- oder 8-Kanal Modul.

8.2.1 Beispiel 1: AND-Link (Zustand mit Flanke)

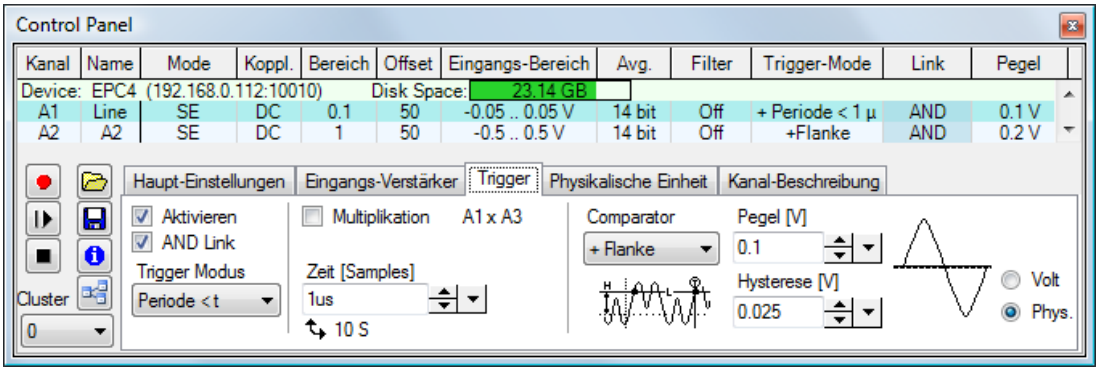
Nur wenn das **Signal "*Line Voltage*"** (rote Kurve) **oberhalb** von **100 V** liegt UND dann die positive Flanke des **Kanales "*Line Current*" 0.5 A** durchfährt wird ein **Trigger ausgelöst**.
Dazu müssen beide Kanäle von derselben Gruppe sein und bei beiden Kanälen muss AND Link eingeschalten sein.


TransAS - TransAS 3 Manual.exp
Datei Auto Sequenz Messung Ansicht Kurvenanzeige Extras Hilfe
Seite 1
Kurvenanzeige 1
Line Current
Line Voltage
Zeit [ms]
Control Panel
Kanal Name Mode Koppl. Bereich Offset Eingangs-Bereich Avg. Filter Trigger-Mode Link Pegel Hyst.
Device: EPC4 (192.168.0.112:10010) Disk Space: 23.13 GB
A1 Line SE DC 0.1 50 -0.05 .. 0.05 V 14 bit Off +Flanke AND 0.5 A 0.1 A
A2 A2 SE DC 10 50 -5 .. 5 V 14 bit Off +Flanke OR 1 V 0.5 V
A3 Line V Diff+ DC 100 50 -50 .. 50 V 14 bit Off State Above AND 100 V 0 V
A4 A4 Diff-
Haupt-Einstellungen Eingangs-Verstärker Trigger Physikalische Einheit Kanal-Beschreibung
Eingangs-Mode Aus Massebez. Differentiell
Koppl. DC AC ICP
Bereich 100 V
Offset 50.0 %
Eingangsbereich -50 .. 50 V
Invertieren
Filter OFF
Kanalname Line Voltage
Cluster
Stopped
TA: -41.023 ms
TB: 123.164 ms
TB-TA: 164.188 ms

In diesem Beispiel sucht Kanal 1 (grüne Kurve) nach Perioden kleiner als 1 µs und Kanal zwei sucht nach einer positiven Flanke. Nachdem beide Bedingungen erfüllt sind, wird der Trigger ausgelöst.

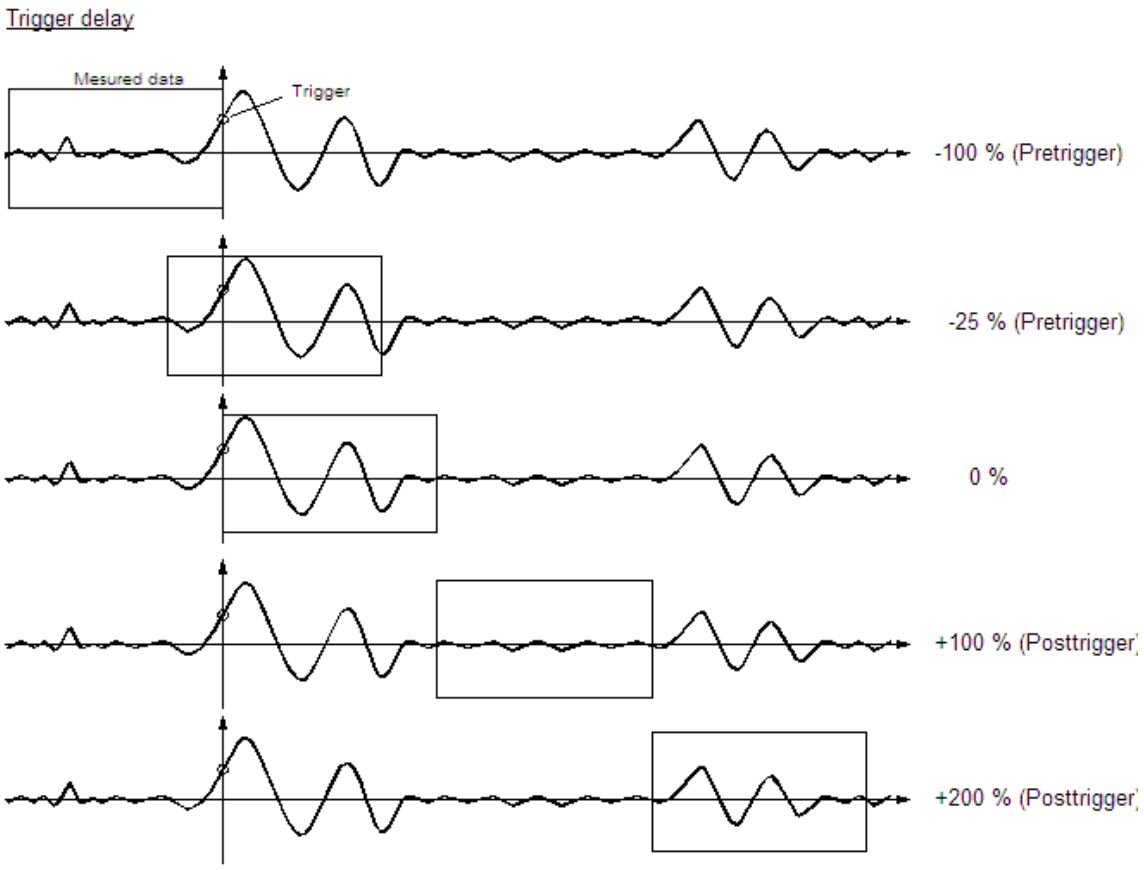
	A1:+Periode<t, t<1µs A2: Flanke, +Flanke

Das Control Panel wiederspiegelt die momentanen Trigger-Einstellungen in den Kolonnen Trigger Mode, Link, Pegel und Hysteresis.



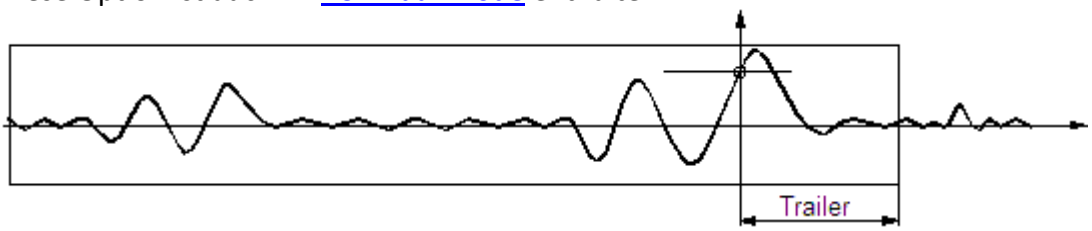
8.3 Pre- und Post-Trigger (Triggerverzögerung)

Diese Datenaufnahme-Parameter werden bei Scope-, Multi-Block- und ECR-Mode verwendet. Die Position des **Aufnahmefensters** (oder Blocks) relativ zum **Triggerzeitpunkt** kann in gewissen Grenzen eingestellt werden. Dabei spricht man von Pre-Trigger, wenn die Aufnahme Teile des Signals vor dem Triggerzeitpunkt enthält. Werden nur Signalanteile nach dem Triggerzeitpunkt aufgezeichnet, spricht man von Post-Trigger. Diese Triggerverzögerung wird in Prozenten der Gesamtaufnahmedauer angegeben. Die TPCX/TPCE-Hardware erlaubt Triggerverzögerungen von -100% bis +200%.



Im kontinuierlichen Mode gibt es keinen Pre- oder Post-Trigger. An dessen Stelle tritt der Stopp-Trigger und der **Nachlauf** (Trailer). Im kontinuierlichen Mode kann die Aufnahme beendet werden, wenn die TPCX/TPCE-Hardware ein Trigger-Ereignis erkennt. Dann spricht man von einem Stopp-Trigger. Manchmal ist es wünschenswert, die Aufnahme dann noch eine Weile fortzusetzen, bevor sie definitiv beendet wird. Dazu dient der Nachlauf-Parameter. Dieser gibt an, wieviele Samples nach dem Stopp-Trigger zusätzlich noch aufgezeichnet werden. Die Date-

nerfassungs-Hardware erlaubt TranAX-Einstellungen von 0 (kein Nachlauf) bis 16 MSamples. Diese Option ist auch im ECR Dual-Mode enthalten.

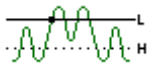


8.4 Trigger-Modi

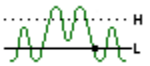
Folgende Trigger-Arten können jeweils für alle triggerfähigen Kanäle individuell eingestellt werden. **"L"** entspricht dem **Trigger-Level** und **H** dem einstellbaren **Hystereseband**.

8.4.1 Flanke

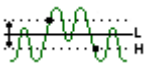
Mit den **Flanken-Einstellungen** kann man entweder die positive, negative oder beide Flanken des Trigger-Signals auswählen. TranAX triggert nur wenn das Signal das **Hystereseband durchlaufen** hat.



\+ Flanke



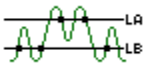
\- Flanke



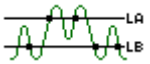
± Flanke

8.4.2 Fenster

Mit der **Fenster-Eintritt** Trigger-Wahl löst TranAX jedes Mal den Trigger aus, falls das Signal in das Fenster eintritt. Entsprechendes gilt für **Fenster-Austritt**.



Fenster-Eintritt

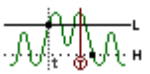


Fenster-Austritt

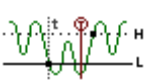
Die folgenden Trigger-Arten benötigen die Option Advanced-Trigger.

8.4.3 Puls > Zeit

Sobald ein positiver oder negativer Puls erkannt wurde, wird ein Trigger-Ereignis nur ausgelöst, wenn der Puls grösser ist als die angegebene Zeit.
Bitte beachten: Um das Ende eines Pulses zu bestimmen, muss die eingestellte Hysterese beachtet werden!



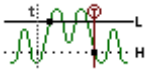
\+ Puls > t



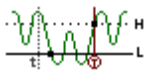
\- Puls > t

8.4.4 Puls < Zeit

Es wird nur getriggert, falls die Pulsweite kleiner ist als die angegebene Zeit.



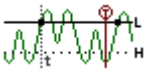
\+ Puls < t



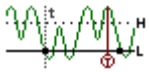
- Puls < t

8.4.5 Periode > Zeit

TranAX triggert nur, wenn eine Periode grösser ist als die eingestellte Zeit. Die Hysterese ermöglicht eine Unterdrückung ungültiger Perioden (z.B. hochfrequentes Rauschen).



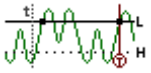
+ Periode > t



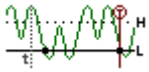
- Periode > t

8.4.6 Periode < Zeit

Wenn eine Periode kürzer als die eingestellte Zeit ist, wird ein Trigger-Ereignis ausgelöst.



+ Periode < t



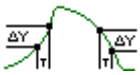
- Periode < t

8.4.7 Flankensteilheit

Mit der Flankensteilheit-Option wird der Trigger bei Überschreiten einer gewissen Anstiegs- oder Abfallsteilheit ausgelöst. Sie wird hauptsächlich gebraucht, um schnelle Parasiten oder Spitzen auf langsamen Signalen zu erkennen.
Die Flankensteilheit kann mittels **Delta Samples** (Delta Zeit) und **Delta-Y** (Delta Amplitude) eingestellt werden. **Delta-Y** sollte immer doppelt so gross sein wie das erwartete Rauschen des Signals. Der **Delta Zeit Parameter ist auf 1024 Samples limitiert**. Die Flankensteilheit-Werte werden auch in der Kanalliste in Kolonne *Trigger Mode* angezeigt.

+ Flankensteilheit

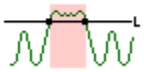
- Flankensteilheit



± Flankensteilheit

8.4.8 Zustand

Der Trigger-Mode **Zustand** steht nur zur Auswahl wenn **AND Link** angewählt ist. Er wird in der Regel in einer UND-Verknüpfung als Qualifizierer verwendet um die Trigger-Auslösung durch andere Quellen von einem Signal-Zustand abhängig zu machen. Die UND-Verknüpfung kann nur innerhalb eines TPCX/TPCE -Modules angewendet werden.



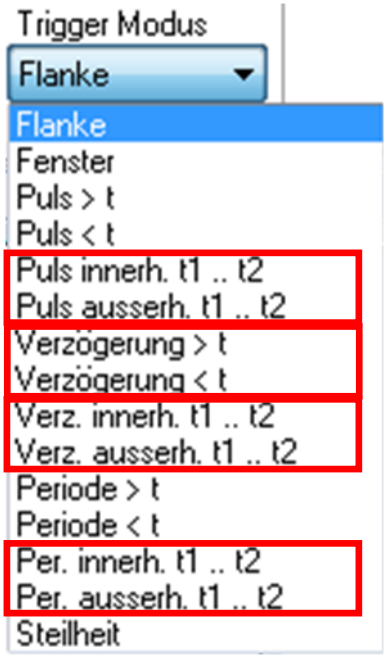
Grösser als

Kleiner als

Ausserdem steht bei einem TPCX/TPCE-Instrument ein **externer Trigger-Eingang** zur Verfügung (TTL). Mit **einem externen Disarm-Eingang** kann die Triggerauslösung durch ein digitales Signal (TTL) verhindert und wieder freigegeben werden. Die diesbezügliche Steckerbelegung findet man im Hardware-Manual.

8.5 Erweiterte Trigger-Funktionen (Übersicht)

Zusätzlich zu den bestehenden sieben Trigger-Modi wurden **acht weitere** implementiert. Diese benötigen ebenfalls die Option *Advanced Trigger*. Auf die bereits vorhandenen Trigger-Modi wird nur am Rande eingegangen.



Die acht neuen Trigger Modi:

- Puls innerhalb t1 .. t2
- Puls ausserhalb t1 .. t2
- Verzögerung > t
- Verzögerung < t
- Verzögerung innerhalb t1 .. t2
- Verzögerung ausserhalb t1 .. t2
- Periode innerhalb t1 .. t2
- Periode ausserhalb t1 .. t2

Im Control-Panel können die Zeiten (t, t1, t2) auch als **Anzahl Samplings** eingegeben werden. Das Programm errechnet damit (multipliziert mit der Abtastrate) die entsprechenden Zeiten. **Intern** werden diese als **Anzahl Samplings** und nicht als Zeiten verwaltet.

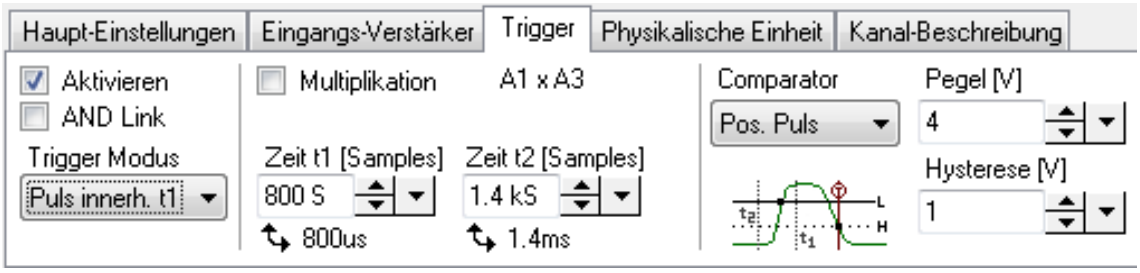
Die eingestellten Zeiten ändern sich, wenn die Zeitbasis verstellt wird!

Um die erweiterten Trigger-Modi zu nutzen, muss die Installierte Software eventuell aktualisiert werden. Die folgenden Versionen werden als **Minimum** vorausgesetzt:

TranAX:	**3.2.1.624**	(Menü ***"Hilfe" / "Info"***)
TPC-Server:	**1.3.2**	(Control Panel /)

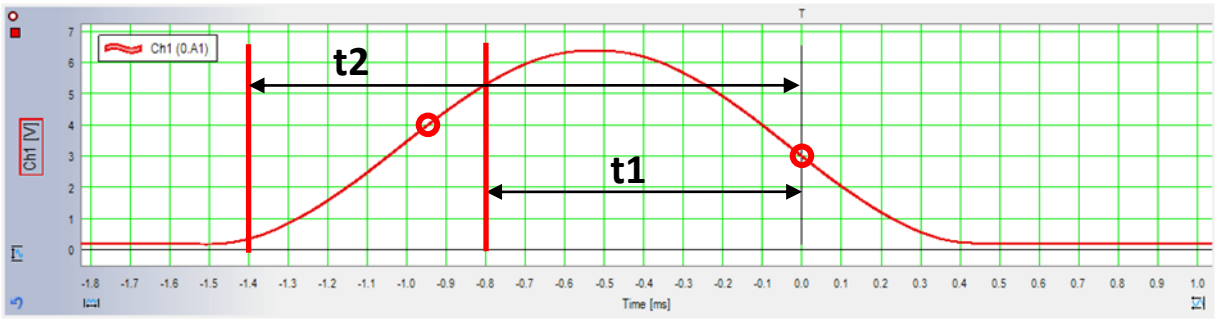
8.5.1 Puls innerhalb t1 .. t2

Der Trigger löst aus, wenn die **Pulsbreite innerhalb** der eingestellten Zeiten **t1 und t2** liegt. In diesem Beispiel sind die Zeiten t1 = 0.8ms und t2 = 1.4ms eingestellt. Abtastrate ist auf 1MS/s eingestellt. 800 Samples entsprechen somit genau 0.8ms, respektive 1.4kS entsprechen den 1.4ms.



Einstellungen für Kanal 1.

Da auf eine positive Flanke getriggert wird (Pos. Puls), ist die Bedingung beim sinkenden Signal erfüllt, der Trigger-Punkt wird auf die sinkende Flanke gesetzt und zwar auf den Wert **Pegel minus Hysterese**. In diesem Beispiel bei 3V.

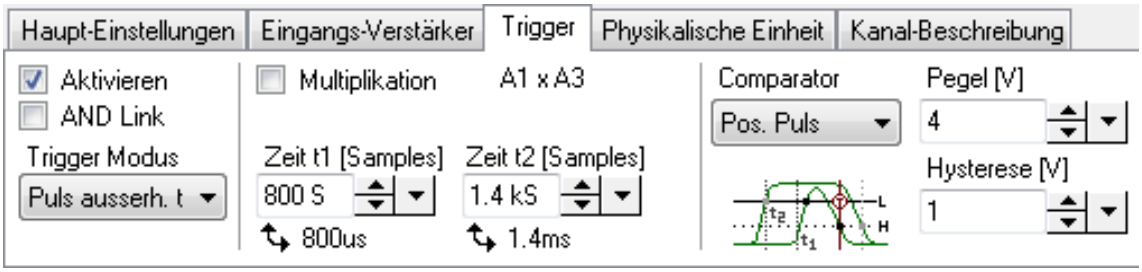


Die Pulsbreite beträgt ca.0.95ms und ist somit innerhalb der beiden Zeiten t1 und t2 von 0.8ms und 1.4ms.

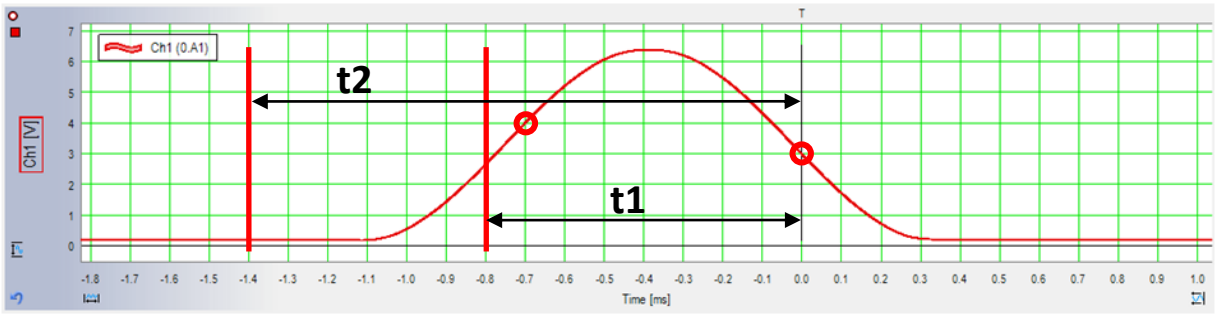
© Elsys AG

8.5.2 Puls ausserhalb t1 .. t2

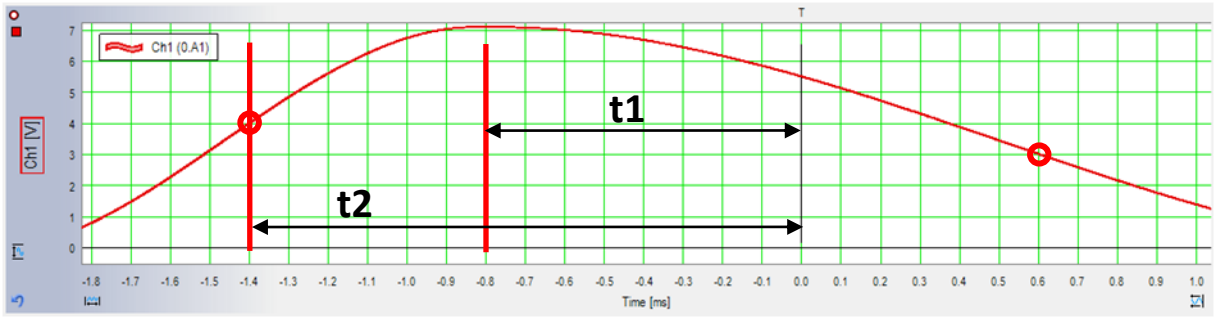
Der Trigger löst aus, wenn sich die Pulsbreite **ausserhalb** der eingestellten Zeiten t1 und t2 befindet. D.h. die Pulsbreite muss kleiner t1 oder grösser t2 sein, damit die Trigger-Bedingung erfüllt ist (Pulsbreite <t1 oder Pulsbreite > t2).



Einstellungen für Kanal 1.



*Hier ist die **Pulsbreite < t1**.*



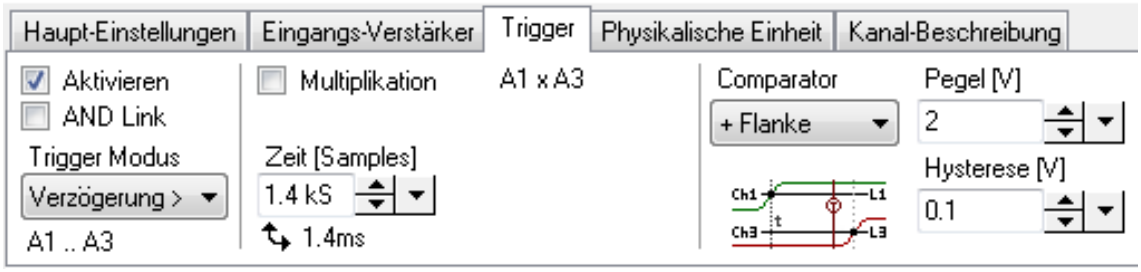
*In diesem Bild ist die **Pulsbreite > t2**.*

Der Trigger-Zeitpunkt befindet sich genau **t2** hinter dem Kriterium für den Beginn des Pulses (hier pos. Flanke, Pegel = 4V). Das Signalverhalten zeigt also genau **beim Trigger-Nullpunkt keinen Trigger-Hinweis**. Das Ende des Pulses (hier neg. Flanke bei 3V) befindet sich dahinter, ev. sogar im Unendlichen oder wird je nach Block-Länge nicht mehr aufgezeichnet.

8.5.3 Verzögerung > t

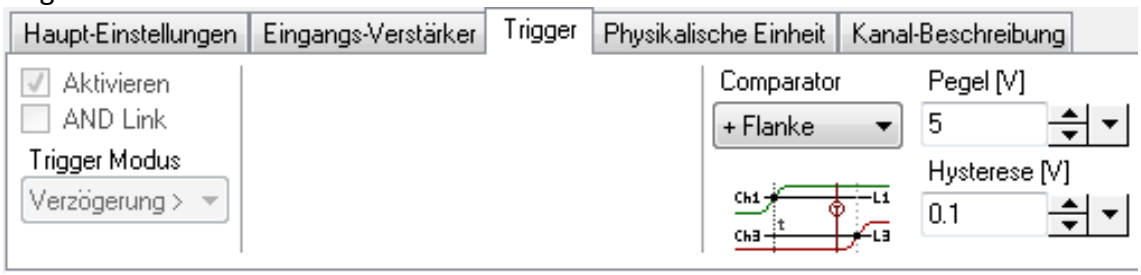
Dieser Trigger-Mode verwendet die Signale der Kanäle 1 und 3 (bzw. 2 und 4). Es wird dabei erfasst, ob die **Zeit** zwischen dem Erfüllen der Trigger-Bedingungen von **Kanal 1** und **Kanal 3 grösser** als die eingestellte **Zeit t** ist. Ist dies der Fall, wird ein Trigger ausgelöst.
Zusätzlich zu der **Zeit t** müssen für die beiden Kanäle 1 und 3 die Einstellungen des Komparators vorgenommen werden (Flanke, Pegel und Hysterese).

Kanal 1 und 3 sowie Kanal 2 und 4 werden für diesen Trigger-Modus kombiniert. Andere Kanal-Kombinationen sind nicht möglich.

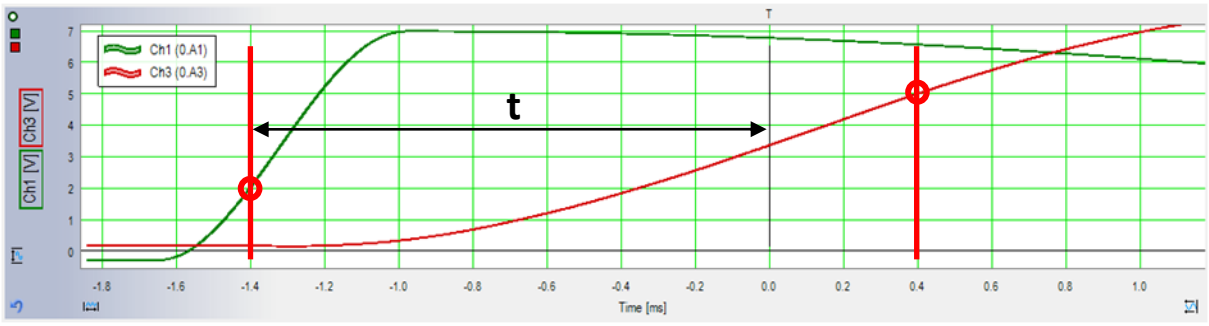


Einstellungen für Kanal 1.

Für den zugehörigen Kanal 3 (bzw. 4) kann keine Änderung am Trigger-Modus vorgenommen werden, dieser wird von Kanal 1 (bzw. 2) bestimmt. Beim zugehörigen Kanal (3 bzw. 4) kann die Flanke unabhängig "+", "-" oder "±" gewählt werden. Ebenso kann der Pegel und die Hysterese eingestellt werden.



Einstellungen für Kanal 3. Der Trigger-Modus ist an jenen von Kanal 1 gekoppelt.



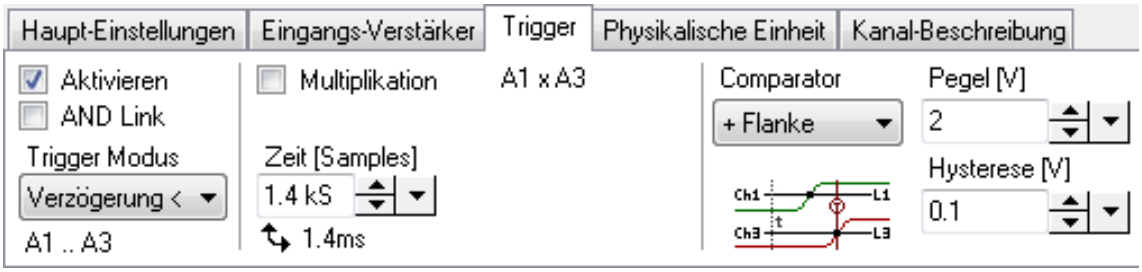
*In diesem Bild ist die **Verzögerung > t**.*

Der Trigger-Nullpunkt befindet sich genau um die **Zeit t** hinter dem Trigger-Kriterium für den ersten Kanal (hier pos. Flanke, 2V). Das Signal-Verhalten zeigt also genau beim Trigger-Nullpunkt keinen Trigger-Hinweis. Das Trigger-Kriterium vom zugehörigen Kanal (hier pos. Flanke, 5V) befindet sich dahinter, ev. sogar im Unendlichen oder wird je nach Block-Länge nicht mehr aufgezeichnet.

© Elsys AG

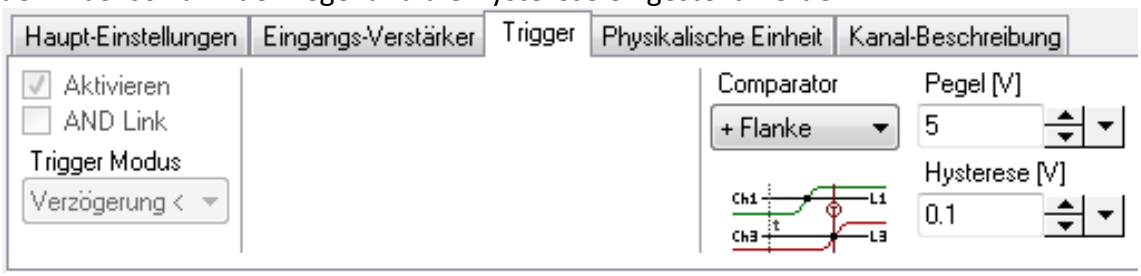
8.5.4 Verzögerung < t

Hier wird ein Trigger ausgelöst, wenn die **Verzögerung kleiner** der eingestellten **Zeit t** ist. Zusätzlich zu der Zeit t müssen für die beiden Kanäle 1 und 3 (bzw. 2 und 4) die Einstellungen des Komparators vorgenommen werden (Flanke, Pegel und Hysterese).

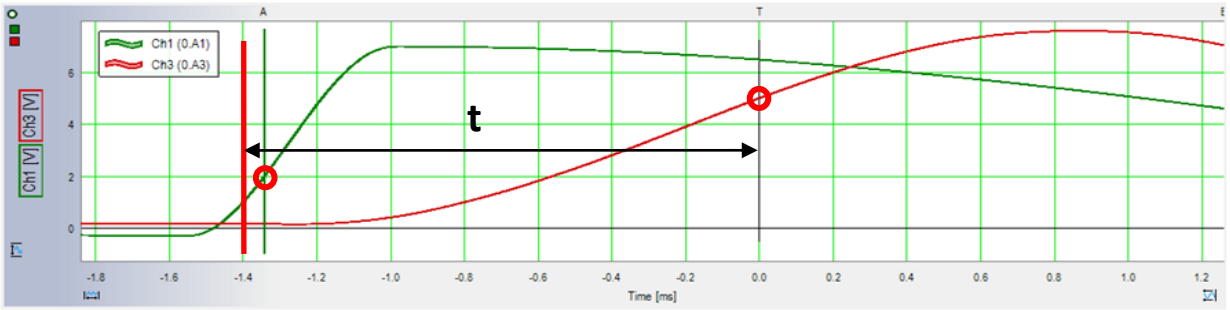


Einstellungen für Kanal 1.

Beim zugehörigen Kanal (3 bzw. 4) kann die Flanke unabhängig "+", "-" oder "±" gewählt werden. Ebenso kann der Pegel und die Hysterese eingestellt werden.



Einstellungen für Kanal 3. Der Trigger-Modus ist an jenen von Kanal 1 gekoppelt.

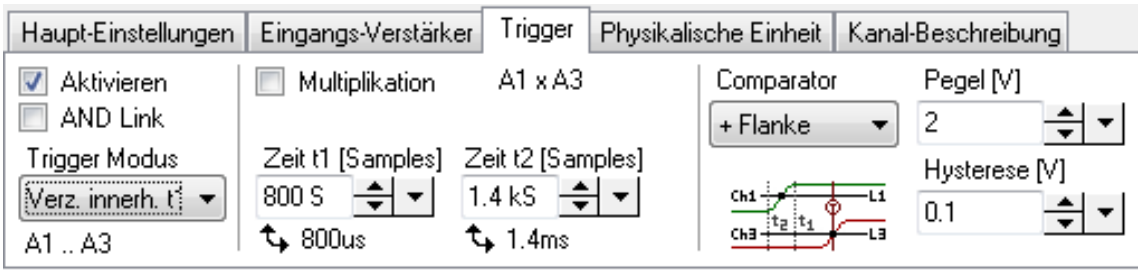


In diesem Bild ist die Verzögerung ***<t.***

8.5.5 Verzögerung innerhalb t1 .. t2

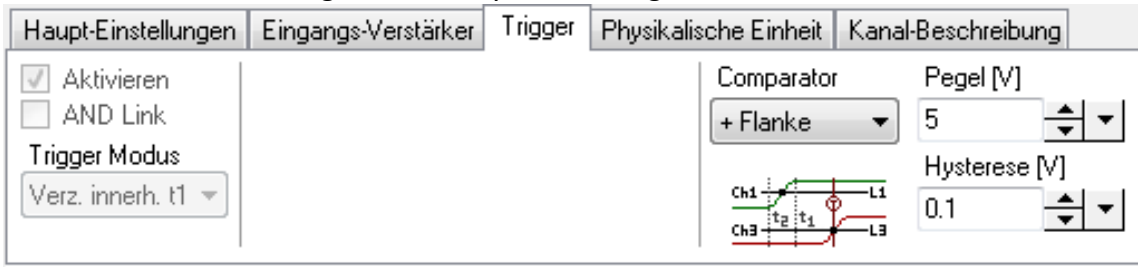
Dieser Trigger-Mode verwendet die Signale der Kanäle 1 und 3 (bzw. 2 und 4). Es wird dabei erfasst, ob die **Zeit** zwischen dem Erfüllen der Trigger-Bedingungen von **Kanal 1** und **Kanal 3 innerhalb** der eingestellten **Zeiten t1 und t2** ist. Ist dies der Fall, wird ein Trigger ausgelöst.

Zusätzlich zu den **Zeiten t1 und t2** müssen für die beiden Kanäle 1 und 3 (bzw. 2 und 4) die Einstellungen des Komparators vorgenommen werden (Flanke, Pegel und Hysterese).

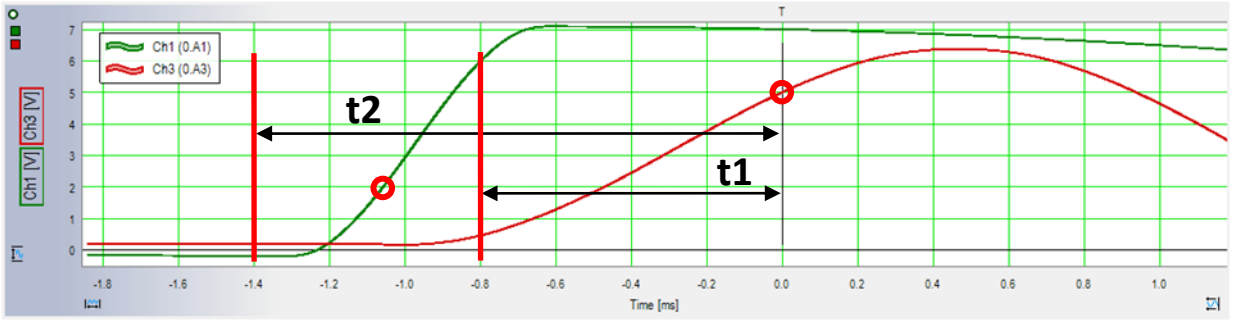


Einstellungen für Kanal 1.

Beim zugehörigen Kanal (3 bzw. 4) kann die Flanke unabhängig "+", "-" oder "±" gewählt werden. Ebenso kann der Pegel und die Hysterese eingestellt werden.



Einstellungen für Kanal 3. Der Trigger-Modus ist an jenen von Kanal 1 gekoppelt.

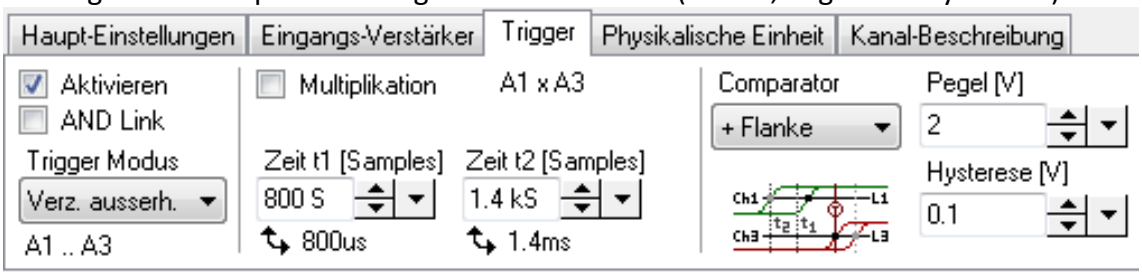


In diesem Bild ist die Verzögerung ca. 1.1ms (d.h. innerhalb von 0.8.und 1.4ms).

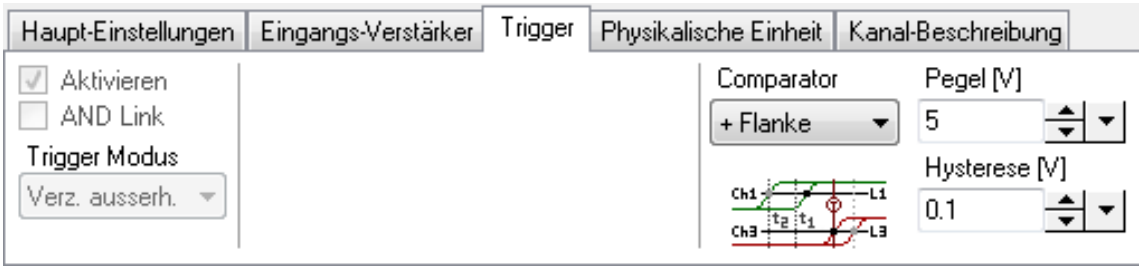
8.5.6 Verzögerung ausserhalb t1 .. t2

Dieser Trigger-Mode verwendet die Signale der Kanäle 1 und 3 (bzw. 2 und 4). Es wird dabei erfasst, ob die **Zeit** zwischen dem Erfüllen der Trigger-Bedingungen von **Kanal 1** und **Kanal 3 ausserhalb** der eingestellten **Zeiten t1 und t2** ist. Ist dies der Fall, wird ein Trigger ausgelöst.

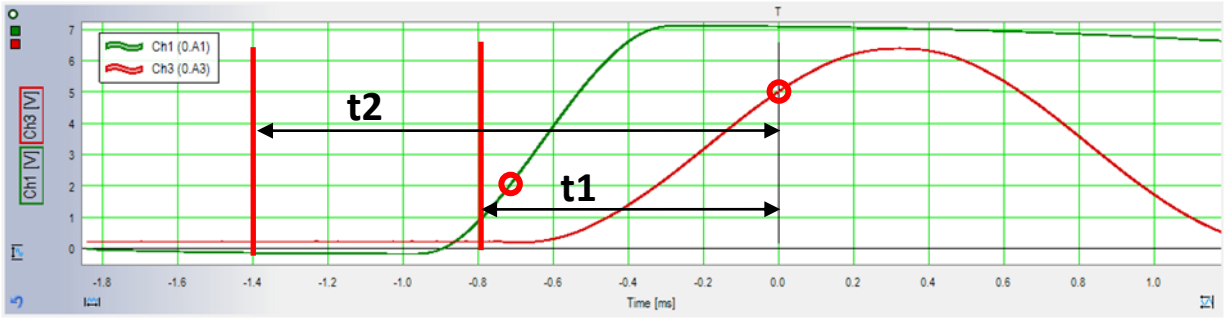
Zusätzlich zu den **Zeiten t1 und t2** müssen für die beiden Kanäle 1 und 3 (bzw. 2 und 4) die Einstellungen des Komparators vorgenommen werden (Flanke, Pegel und Hysterese).



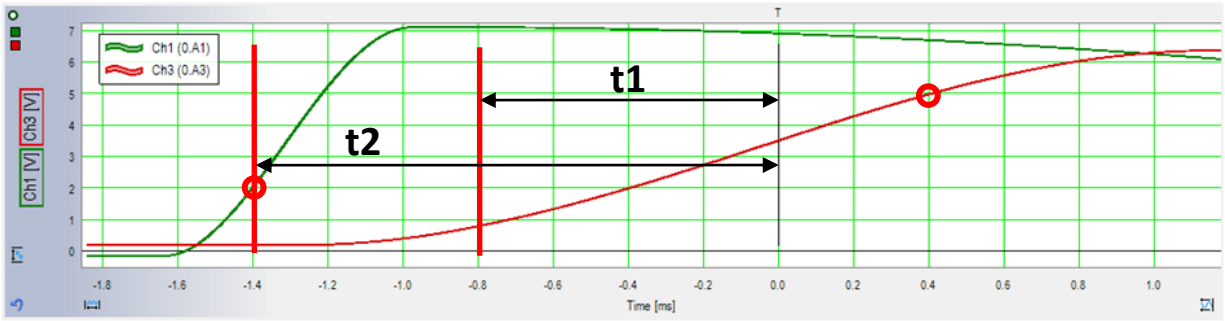
Einstellungen für Kanal 1.



Einstellungen für Kanal 3. Der Trigger-Modus ist an jenen von Kanal 1 gekoppelt.



*In diesem Bild ist die **Verzögerung < t1***

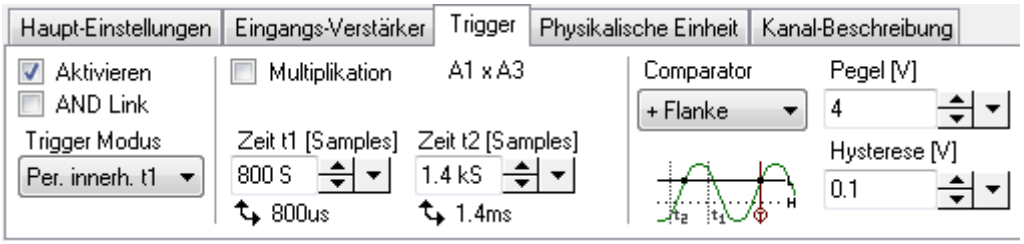


*In diesem Bild ist die **Verzögerung >t2**.*

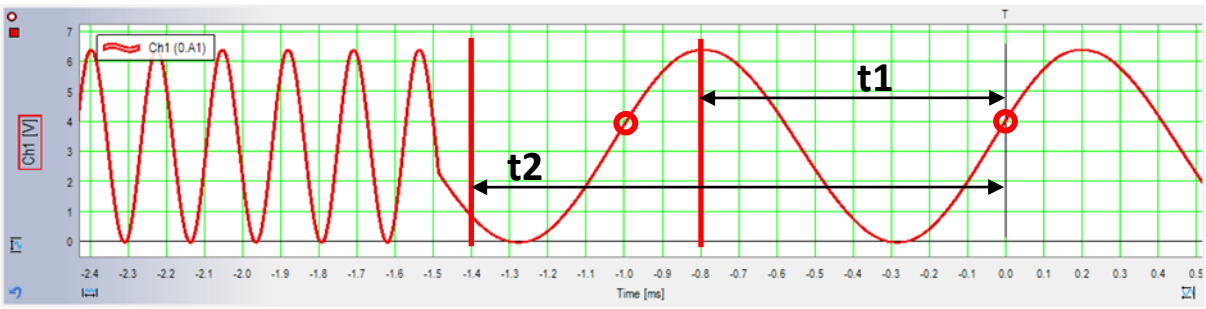
Der Trigger-Zeitpunkt befindet sich genau t2 hinter dem Trigger-Kriterium für den ersten Kanal (hier pos. Flanke, Pegel = 2V). Das Signal-Verhalten zeigt also genau beim Trigger-Nullpunkt keinen Trigger-Hinweis. Das Trigger-Kriterium für den zugehörigen Kanal (hier pos. Flanke, Pegel = 5V) befindet sich dahinter, ev. sogar im Unendlichen oder wird je nach Block-Länge nicht mehr aufgezeichnet.

8.5.7 Periode innerhalb t1 .. t2

Ein Trigger-Event wird ausgelöst, sobald die **Periodendauer** sich **innerhalb** der beiden Zeiten **t1 und t2** befindet.



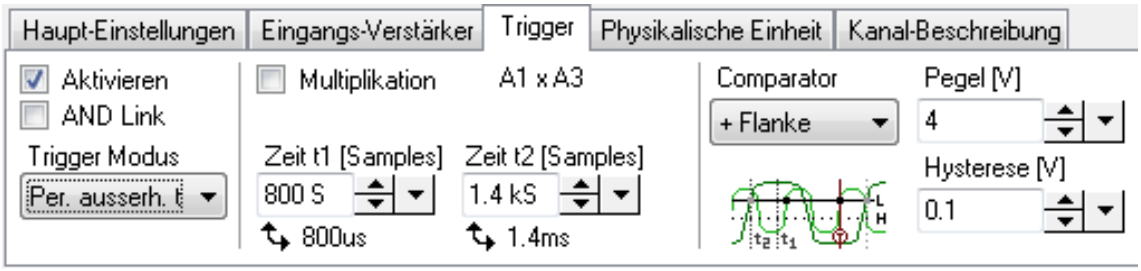
Einstellungen für Kanal 1.



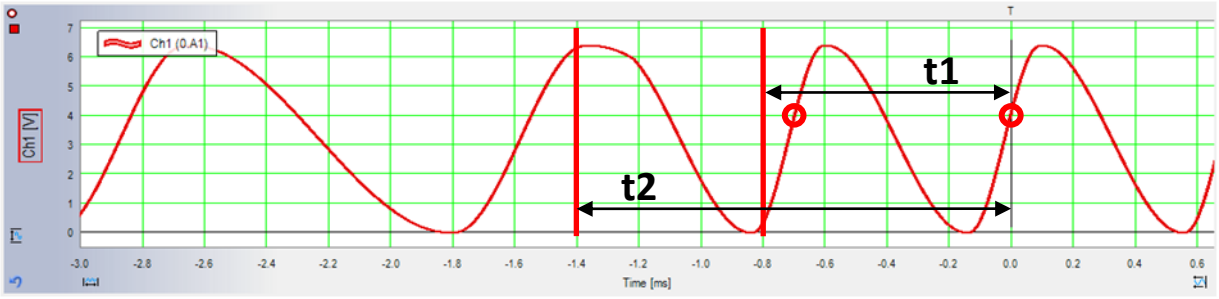
In diesem Bild ist die Periodendauer ca.1.0ms (d.h. innerhalb von 0.8.und 1.4ms).

8.5.8 Periode ausserhalb t1 .. t2

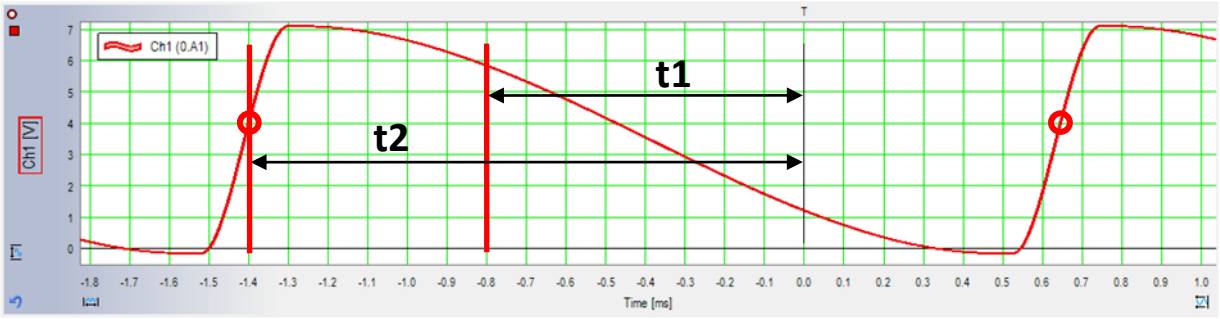
Ein Trigger-Event wird ausgelöst, wenn sich die **Periodendauer ausserhalb** der beiden Zeiten **t1** und **t2** befindet



Einstellungen für Kanal 1.



*In diesem Bild ist die Periodendauer **<t1**.*

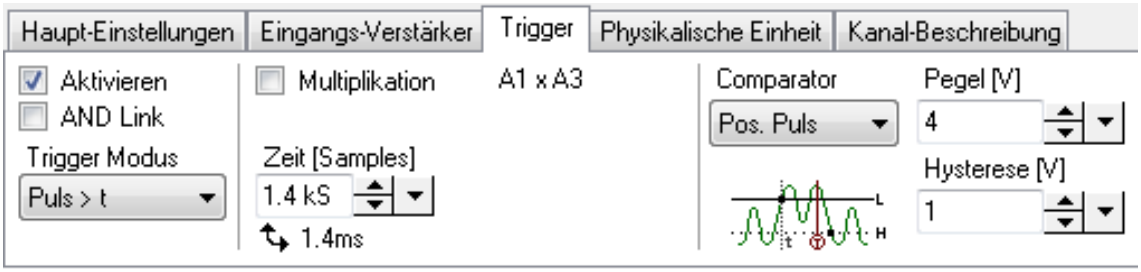


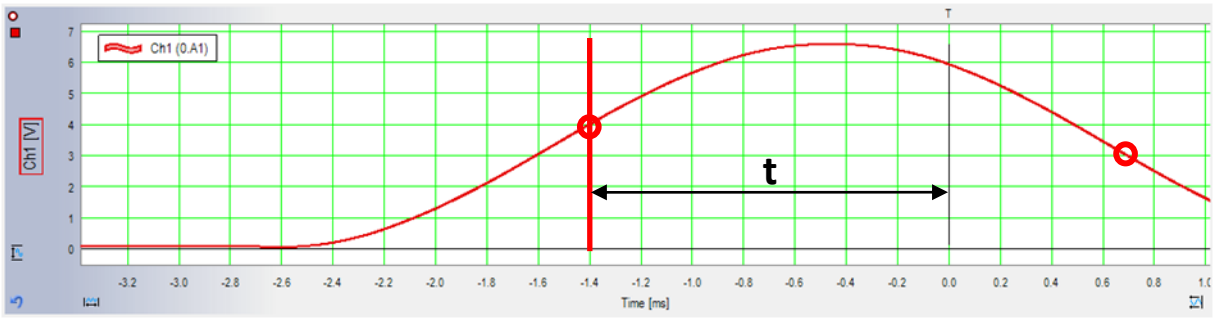
*In diesem Bild ist die Periodendauer **>t2**.*

Der Trigger-Zeitpunkt befindet sich genau **t2** hinter dem Kriterium für den Beginn einer Periode (hier pos. Flanke, Pegel = 4V). Das Signal-Verhalten zeigt also genau beim Trigger-Nullpunkt keinen Trigger-Hinweis. Das Ende der Periode (hier pos. Flanke bei 4V) befindet sich dahinter, eventuell. sogar im Unendlichen oder wird je nach Block-Länge nicht mehr aufgezeichnet.

8.6 Bisherige Trigger-Modi bezogen auf Puls/Periode

8.6.1 Puls > t

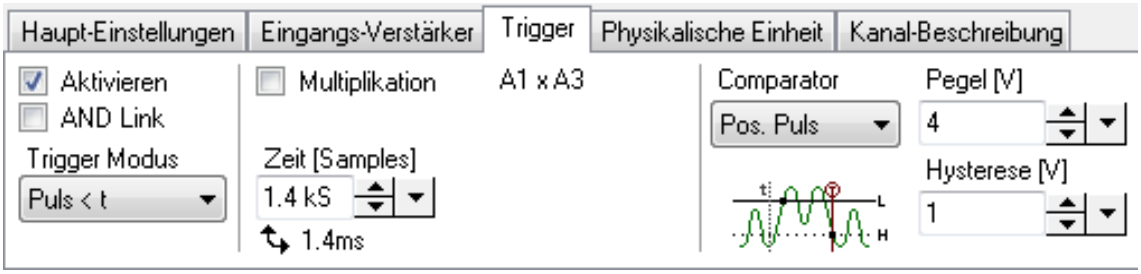


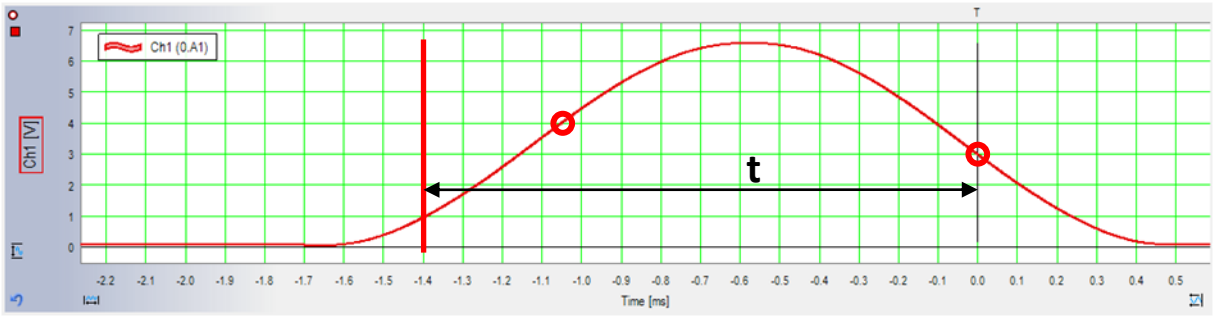


*In diesem Bild ist die **Pulsbreite > t**.*

Der Trigger-Zeitpunkt befindet sich genau um die Zeit **t** hinter dem Kriterium für den Beginn des Pulses (hier pos. Flanke, Pegel = 4V). Das Signal-Verhalten zeigt also genau beim Trigger-Nullpunkt keinen Trigger-Hinweis. Das Ende des Pulses (hier neg. Flanke bei 3V) befindet sich dahinter, ev. sogar im Unendlichen oder wird je nach Block-Länge nicht mehr aufgezeichnet.

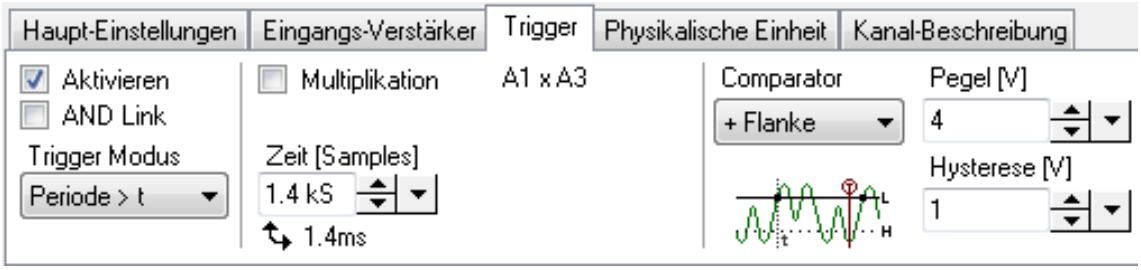
8.6.2 Puls < t

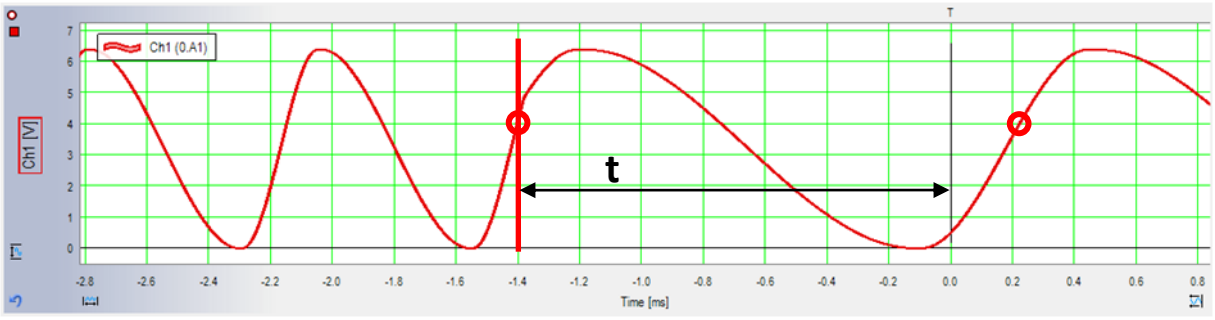




*In diesem Bild ist die **Pulsdauer < t**.*

8.6.3 Periode > t

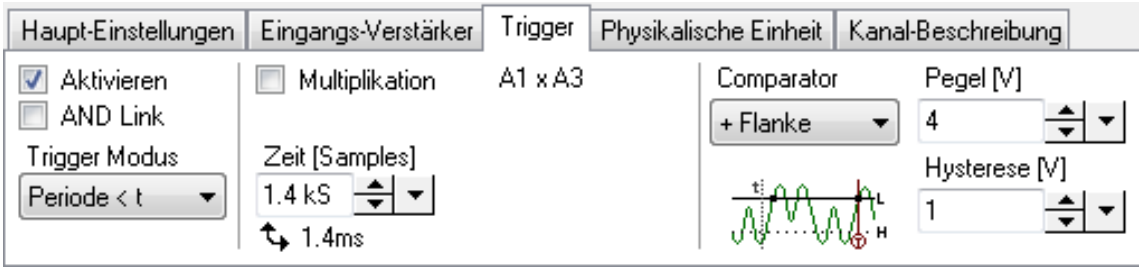


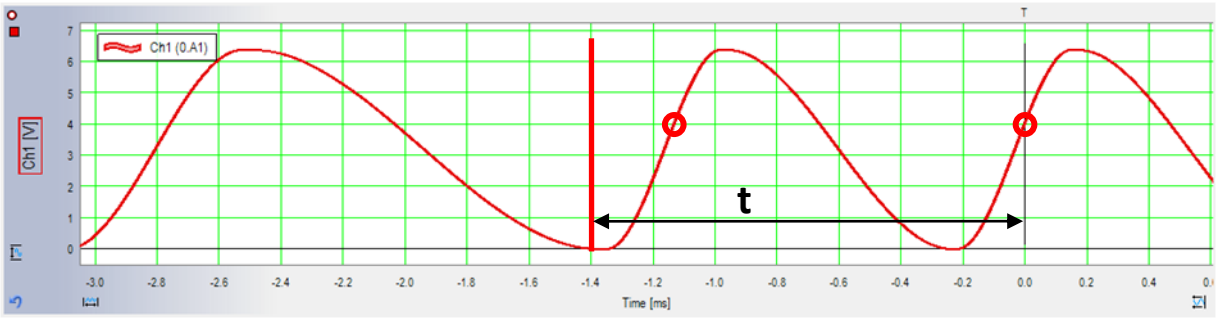


*In diesem Bild ist die Periodendauer **> t**.*

Der Trigger-Zeitpunkt befindet sich genau um die Zeit **t** hinter dem Kriterium für den Beginn einer Periode (hier pos. Flanke, Pegel = 4V). Das Signal-Verhalten zeigt also genau beim Trigger-Nullpunkt keinen Trigger-Hinweis. Das Ende der Periode (hier pos. Flanke bei 4V) befindet sich dahinter, ev. sogar im Unendlichen oder wird je nach Block-Länge nicht mehr aufgezeichnet.

8.6.4 Periode < t

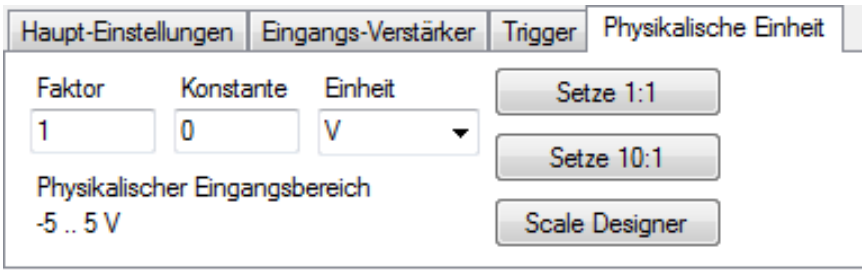




*In diesem Bild ist die Periodendauer **< t**.*

9 Physikalische Einheit

TranAX kann die **gemessenen Kurven** in eine frei wählbare **physikalische Einheit umrechnen** (lineare Transformation).



In dieser Registerkarte können für jeden Kanal individuell eingestellt werden:

- **Faktor und Konstante** für die Umrechnung
- die **physikalische Einheit**

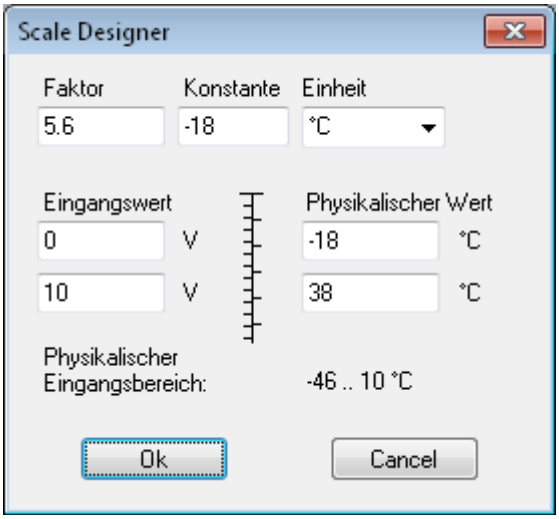
Die Umrechnungsformel lautet:

Physikalische Grösse = (Gemessener Wert [V] * Faktor) + Konstante [Physikalische Einheit]

Die Einstellungen für die Umrechnung müssen vor der Aufnahme eingegeben werden.

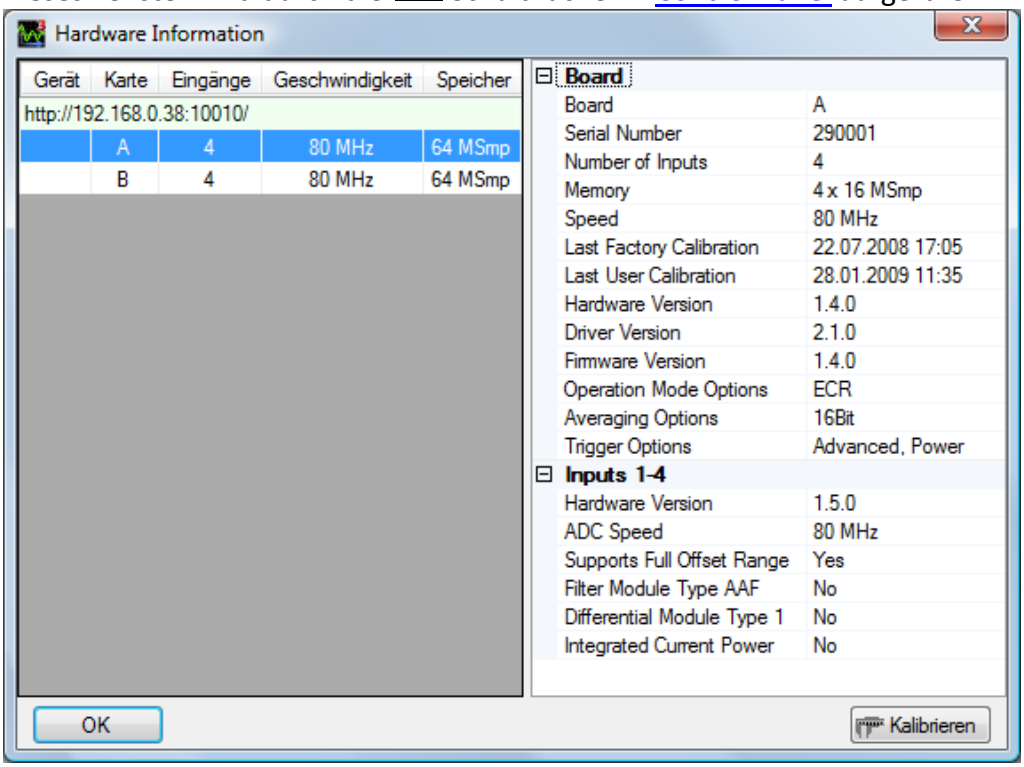
9.1 Scale Designer

Um den Faktor und die Konstante **komfortabel** aus **zwei Referenzpunkten** zu ermitteln, kann der **Scale Designer** verwendet werden, welcher mit der entsprechenden Schaltfläche aufgerufen werden kann.



10 Informationsfenster

Dieses Fenster wird durch die Schaltfläche im Control Panel aufgerufen.



Es zeigt die **aktuell verwendete Hardware** an, mit der TranAX gerade verbunden ist. Diese kann entweder eine oder mehrere auf dem lokalen Rechner eingebauten TPCX/TPCE-Module sein, oder auch eine externes Gerät wie ein TraNET EPC oder TraNET FE. Im Falle eines Fehlers wird eine entsprechende Meldung angezeigt.

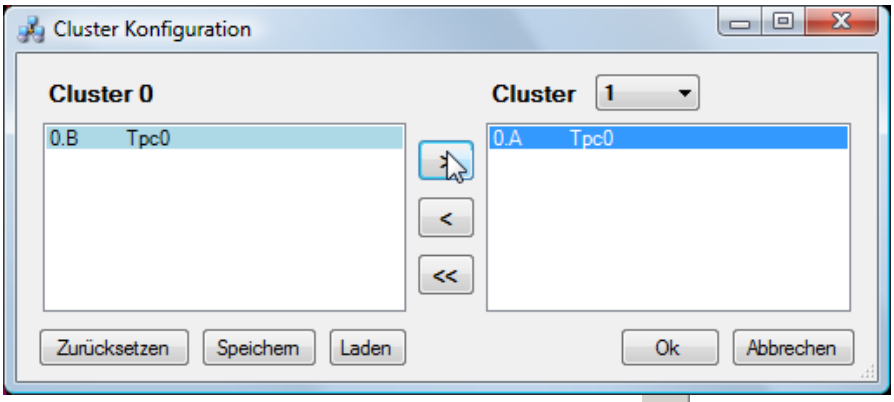
Kalibrieren Um die Mess-Hardware zu kalibrieren, kann die Kalibrations-Schaltfläche gedrückt werden. Der Vorgang dauert einige Sekunden.

Die Hardware sollte erst kalibriert werden, wenn diese ihre **Betriebstemperatur** erreicht hat. Eine Kalibration im kalten Zustand kann zu späteren Ungenauigkeiten bei den Messungen führen!
Um die Kalibration zu starten, müssen laufende Aufnahmen erst gestoppt werden.

Die Kalibration kann jederzeit wiederholt werden.

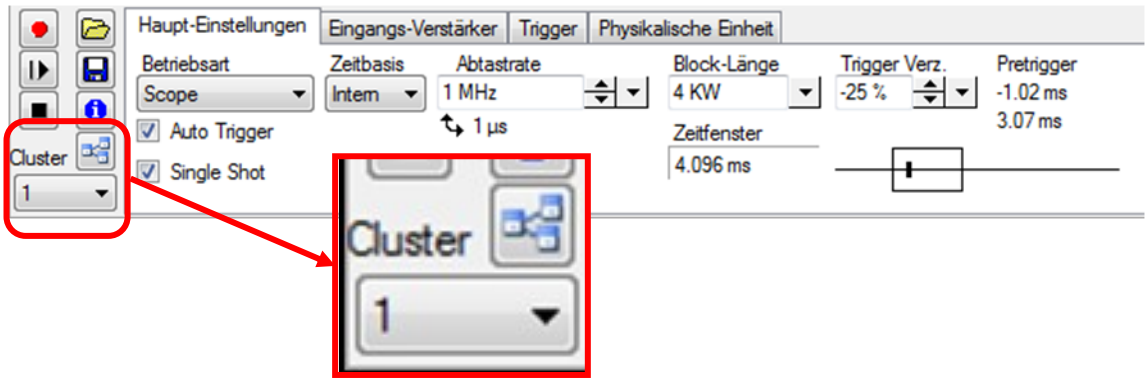
11 Cluster Konfiguration

Die Clusters dienen dazu, **Gruppen von Datenerfassungs-Kanälen zu definieren**, die mit der gleichen Konfiguration arbeiten. Mit Hilfe der Clusterung können unterschiedliche Abtastraten und Betriebsarten eingestellt werden.



Das Cluster Konfigurationsfenster wird durch die Schaltfläche im Control Panel aufgerufen.

Es können verschiedene TPCX/TPCE-Gruppen mit der Cluster Konfigurationsmaske definiert werden. **Anfangs sind alle TPCX**/TPCE-**Module im Cluster 0**. Um ein Modul zu einem Cluster hinzuzufügen, wählt man in der rechten oberen Drop-Down Liste die Cluster Nummer welche als Platzhalter verwendet werden soll. Danach wählt man in der linken Hälfte das gewünschte TPCX/TPCE-Board und drückt die > Schaltfläche, um dieses Modul dem gewählten Cluster hinzuzufügen.
Nachdem die Cluster definiert wurden, können sie via Drop-Down Liste links von den Einstell-Registerkarten ausgewählt werden:



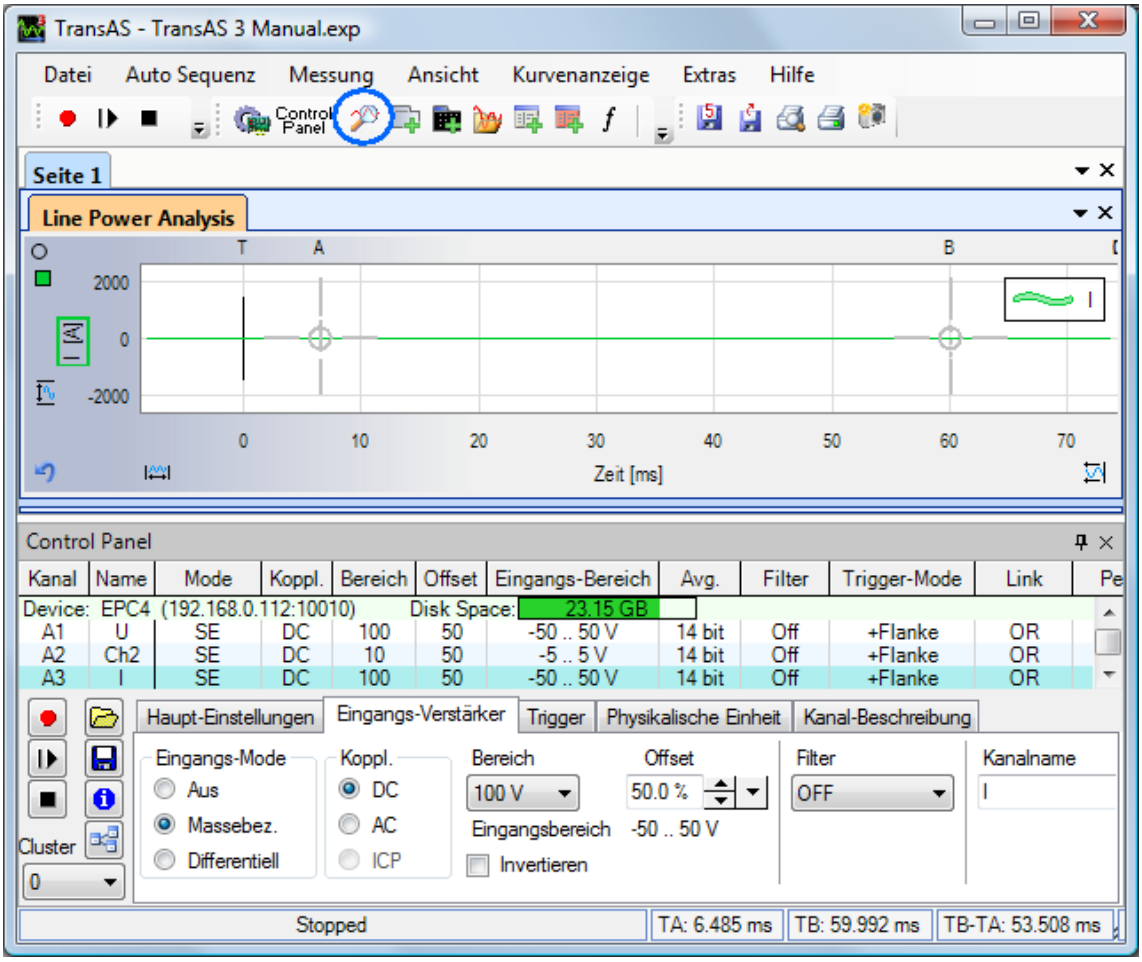
Jeder Cluster hat seine eigene Betriebsart und individuelle Einstellungen (Abtastrate, Block Länge, Triggerverzögerung etc.)

Alle Cluster arbeiten **synchron** betreffend **Systemtakt** und **Triggerlogik**. Falls ein Cluster ein Trigger-Ereignis auslöst, werden **alle anderen Cluster** auch auf dieses **Ereignis reagieren**, egal in welcher Betriebsart sie sich befinden.
Darauf sollte geachtet werden, wenn mit verschiedenen Betriebsarten gearbeitet wird (z.B. würde ein Trigger im Single-Shot-Cluster auch den kontinuierlicher-Mode mit Stop-Trigger beenden).

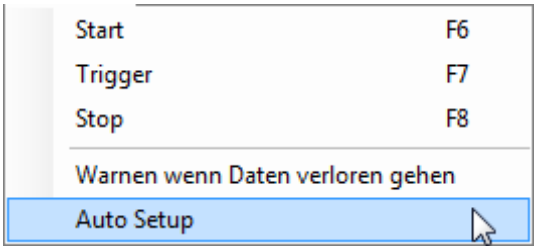
 © Elsys AG

12 Autosetup

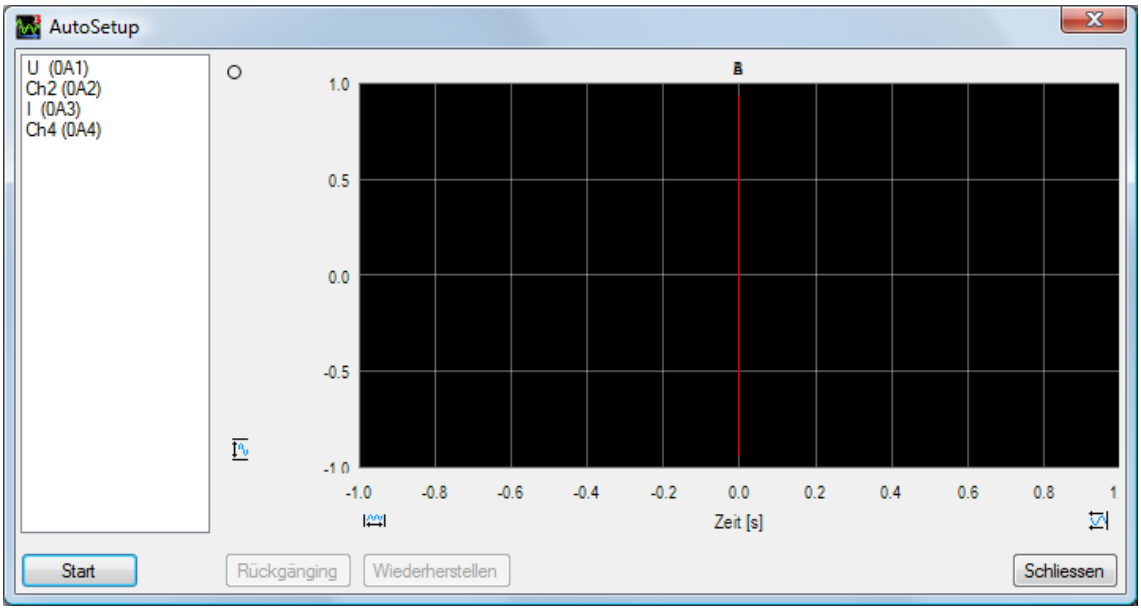
Die Funktion Autosetup kann behilflich sein, falls ein **unbekanntes Signal** an den Transientenrekorder angeschlossen wird und ein **schnelles Einstellen** erwünscht ist.
Die Funktion Autosetup passt den **vertikalen Eingangsbereich** an das Signal an, stellt die **Abtastrate auf das Maximum** und die **Blocklänge auf ca. 20 ms**.
Im folgenden Bild wurde ein Signal im 100 V Eingangsbereich erfasst.



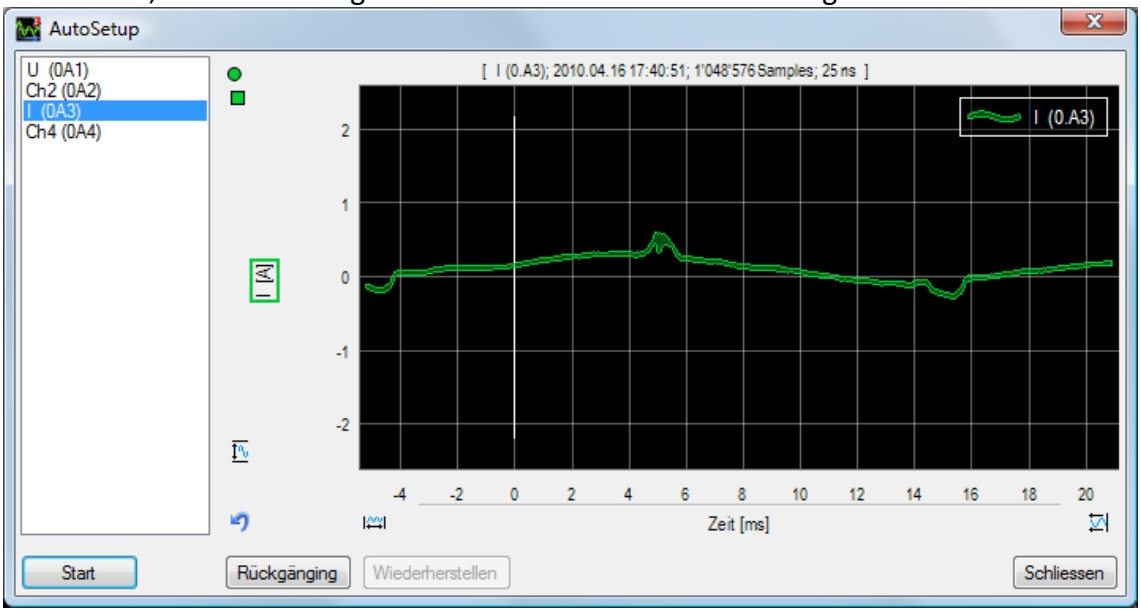
Um den Eingangsbereich schnell zu finden und eine Erfassung auszuführen, ohne die Triggerbedingungen zu kennen, kann via Klick auf das Icon oder im Menü ***"Messung" / "Auto Setup"*** eine automatische Bereichseinstellung vorgenommen werden.



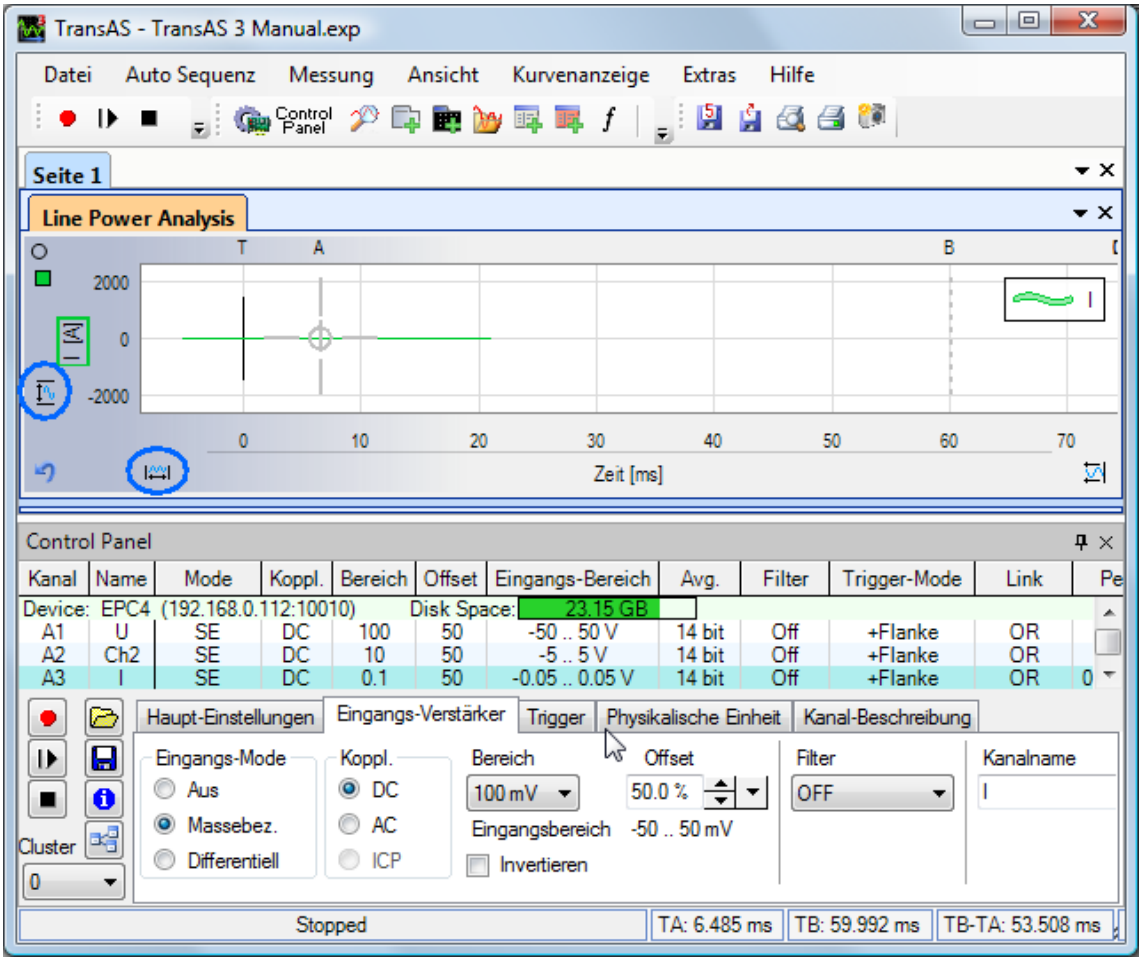
Das **Fenster AutoSetup** erlaubt nun alle zur Verfügung stehenden Kanäle oder einzelne davon für die Autosetup-Funktion zu selektieren.



Ein **einzelner Kanal** wird mit einem **Links-Klick** selektiert. Mehrere Kanäle können mit einem **Ctrl-Links-Klick** angewählt oder ein- und ausgeschalten werden. Sobald der Knopf "Start" gedrückt wird, werden die angewählten Kanäle der Reihe nach abgearbeitet.



Nachdem Autosetup durchgeführt wurde, wird die Kurve in dem neu ermittelten Messbereich dargestellt. An dieser Stelle, und auch beim erneuten Öffnen dieses Fensters, kann mit einem Klick auf **Rückgängig** der **alte Zustand wiederhergestellt** werden. Falls die neuen Einstellungen übernommen werden sollen, so wird dies mit einem Klick auf Schliessen ausgeführt.

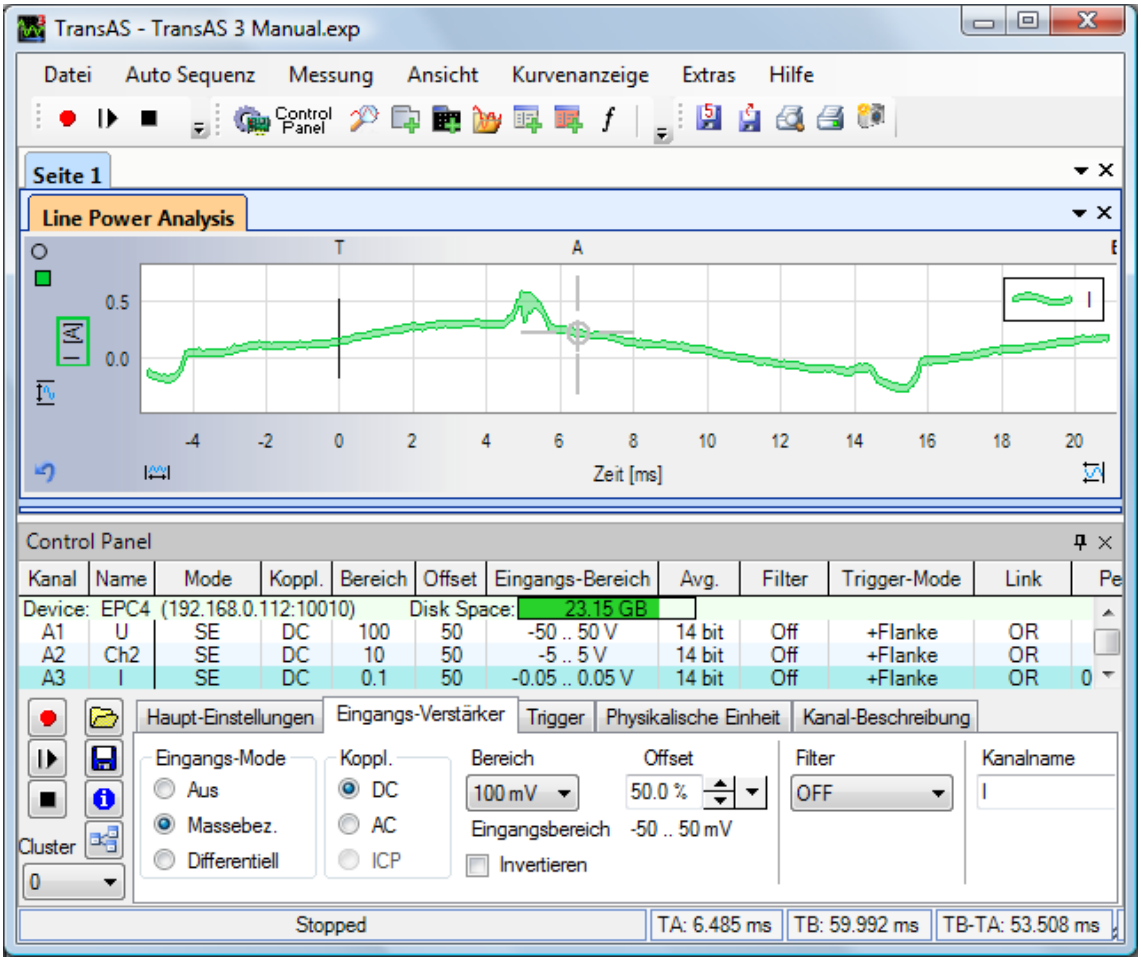


Der Eingangsbereich wurde von 100 V auf 100 mV geändert, die Kurve selbst wird mit derselben Ansicht dargestellt wie zuvor. **Die Kurvenanzeige wird nicht automatisch neu skaliert.**

Durch klicken der beiden Knöpfe **Volle Ansicht X** und **Volle Ansicht Y kann dies nun manuell durchgeführt werden.**

	Die X-Achse auf volle Ansicht strecken.
	Die Y-Achse auf volle Ansicht strecken.

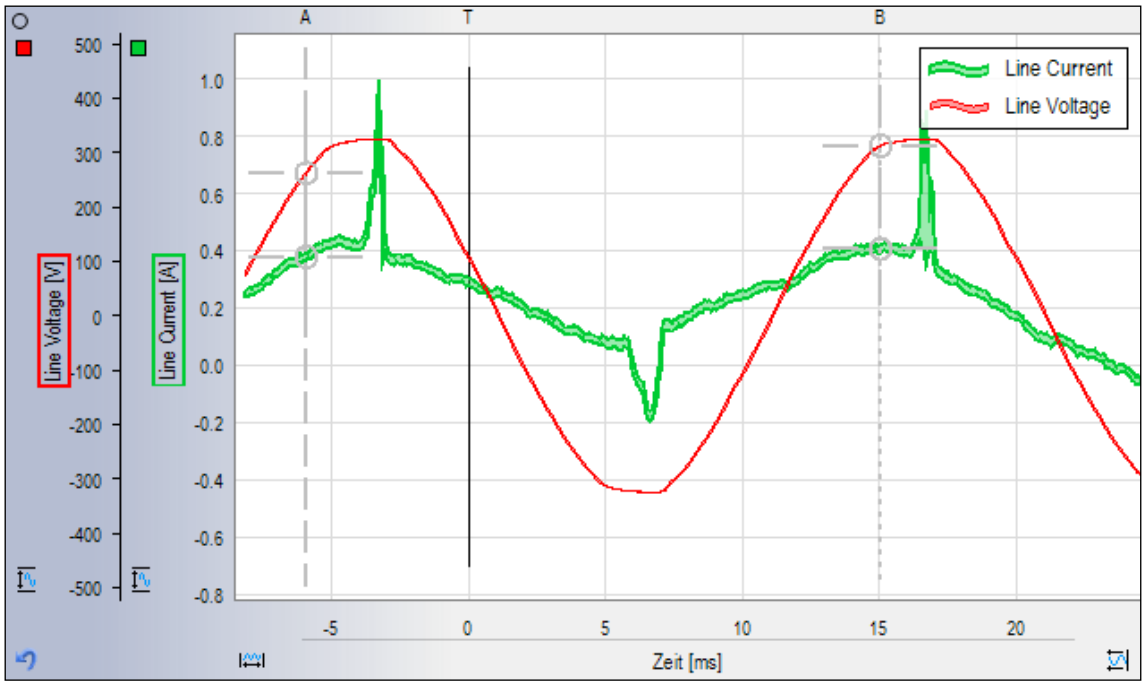
Nachdem die beiden oben erwähnten Zoom-Knöpfe gedrückt wurden, kann man die Details der Kurve erkennen. Die Ansicht kann jederzeit angepasst werden, in dem zum Beispiel näher in die Kurve gezoomt wird.



Mit besserer Kenntnis über das Signal können nun die Trigger-, Vertikal- und Zeitbasis-Einstellungen gezielter vorgenommen werden.
Falls die Triggerquelle eines angewählten Kanales eingeschalten ist, so wird deren Trigger-Mode automatisch auf positive Flanke gestellt und der Pegel eingemittet. Ansonsten werden die Triggereinstellungen nicht verändert.

13 **Kurvenanzeige** (Allgemeines)

Die Kurvenanzeige dient der Visualisierung von aufgenommenen oder laufenden Signale.

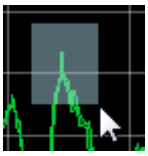


13.1 Organisieren und Platzieren

Mittels Drag & Drop kann ein Kanal vom Control Panel oder vom Signalquellen-Browser in der Kurvenanzeige dargestellt werden. Falls mehrere Signale in einer Kurvenanzeige überlagert werden, erhält jedes Signal **eine bestimmte Farbe** und wird am linken Rand durch ein **farbiges Kästchen repräsentiert**. Jede Kurvenanzeige gehört zu einer Seite. Eine **neue Kurvenanzeigen** wird mittels der Schaltfläche oder im Menü ***"Ansicht" / "Neue Kurvenanzeige"*** hinzugefügt. Um die Kurvenanzeigen neu zu platzieren geht man gleich vor wie mit den Sub-Fenstern: Einfach auswählen und an die gewünschte Stelle verschieben.

Um eine **bessere Übersicht** zu wahren, falls viele Seiten und Anzeigen offen sind, können kann mittels **Rechtsklick** auf eine Seite- oder Kurven-Registerkarte das **Kontextmenü** geöffnet werden. Hier kann man dann die Anzeigen schliessen, einen Titel anpassen oder die Fenster vertikal oder horizontal platzieren.

13.2 Zoomen

	Man kann einen **Bereich vergrössern,** indem ein **Rechteck** über den gewünschten Bereich gezogen wird. Dazu die obere linke Ecke anwählen und mit gedrückter linker Maustaste bis zur unteren rechten Ecke ziehen. Anschliessend die Maustaste wieder loslassen.
	Den Mauszeiger auf die Achsenbeschriftung bewegen bis ein **Zoom-Symbol** erscheint. Anschliessend die linke Maustaste drücken und je nachdem, ob die Maus nach oben/unten bzw. rechts/links bewegt wird, vergrössert oder verkleinert sich der Ausschnitt. Hier kann auch mit dem Scrollrad gezoomt werden.

13.3 Kurven verschieben

Um den Ausschnitt in der Kurvenanzeige zu verschieben, kann innerhalb des Kurvenfensters an einer freien Stelle die **rechte Maustaste gedrückt** und in die gewünschte Richtung bewegt werden.
Andererseits kann der Mauszeiger über die Achseneinheiten platziert werden bis ein doppelseitiger Pfeil erscheint. Danach kann durch Klicken und Bewegen die X- bzw. Y-Achse mitsamt den Kurven verschoben werden. Auch hier kann mit dem Scrollrad gearbeitet werden.

Verschieben des Kurven-Zeitbereichs (X-Skala) mit dem Maus-Rad geht auch, wenn der Maus-Cursor innerhalb des Kurvenfensters ist und gleichzeitig die Shift-Taste gedrückt wird.

13.4 Volle Ansicht

Es gibt zwei Icons um **die volle Ansicht** wieder herzustellen:

	Zeige Y-Achse in voller Ansicht (für jede Achse einzeln).
	Zeige X-Achse in voller Ansicht.

Mit folgenden Schaltern können Sie **Zoomen und Verschieben rückgängig** machen oder wiederherstellen.

	Letzte Änderung rückgängig machen (Undo).
	Änderung wiederherstellen (Redo).
	Falls im Kontinuierlichen oder ECR Mode **Autoscroll** aktiviert ist, wird die Kurvenanzeige automatisch mit dem Signal **horizontal mitlaufen.** Falls Deaktiviert, wird es die Anzeige belassen, aber die Aufnahme läuft weiter!

© Elsys AG

14 Y/T-Kurven-Anzeige-Fenster

Die folgenden Anzeige-Optionen werden mittels **Rechtsklick** im **Kurvenanzeige-Fenster** oder (Auszugsweise) via Menü ***"Kurvenanzeige"*** erreicht.

Verschieden Parameter, wie z.B. Cursors, Gitter, Triggerlinien usw. können zu oder abgeschaltet werden.

Cursor A, B (oder ev. zusätzliche) können in der Kurvenanzige an der aktuellen Position des Maus-Zeigers platziert werden. Natürlich muss dazu "Cursors" aktiviert sein.

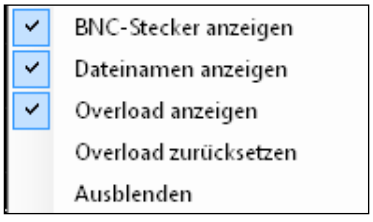
Das Gleiche gilt für die Horizontal-Cursors.

Die Anzahl der Achsen (links und rechts) können auch via Rechts-Klick auf den Y-Achsen selbst definiert werden.

Via "Eintrag hinzufügen" lassen sich (auch nachträglich) an der aktuellen Maus-Position RecordingLog-Einträge vornehmen.
Über "Gehe zu Eintrag" wird der Kurvenanzeige-Ausschnitt direkt beim entsprechenden RecordingLog-Eintrag gesetzt.

14.1 Legende

Die Legende kann ein- oder ausgeblendet werden (Ausblenden). Falls diese nicht mehr sichtbar ist, kann sie jederzeit über das Menü ***"Kurvenanzeige" / "Legende"*** wieder angezeigt werden.

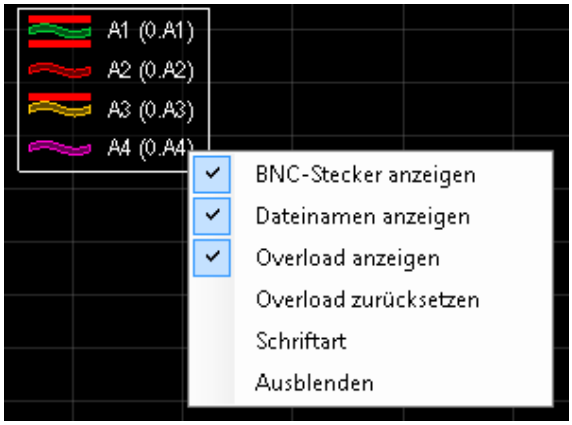


Ein Rechtsklick zeigt das Pop-up-Menü auf der linken Seite. Um die Legende an eine andere Stelle zu verschieben, fahren Sie mit der Maus über die Legende und halten Sie die linke Maustaste gedrückt.

Falls **BNC-Stecker** nicht angewählt ist, wird nur der Kanalname angezeigt. Falls eine Kurve von einer Datei importiert oder mit dem Formel-Editor berechnet wurde kann der **Dateiname** eingeblendet werden.

Die **Option "Overload anzeigen"** zeigt an, ob der Verstärker des entsprechenden Kanals positiv oder negativ übersteuert wird oder wurde. Dies wird mit einer Linie oberhalb bzw. unterhalb des Kurven-Symbols dargestellt.

Die Überlauf-Anzeige signalisiert mit einer roten Linie eine momentane Übersteuerung. Eine gelbe Linie signalisiert eine Übersteuerung die vorgängig (seit Start der Aufzeichnung) mindestens einmal aufgetreten ist.
Ein registrierter Überlauf (gelbe Linie) kann zurück gesetzt werden, indem auf "Overload zurücksetzen" geklickt wird.



Durch einen Klick auf den Menüpunkt **"Overload zurücksetzen"** werden die gelben Markierungen, welche den Overload signalisieren, oberhalb oder unterhalb der Kurve wieder entfernt.

Das **Zurücksetzen** eines eingetretenen Overload-Ereignisses kann sinnvoll sein, wenn z.B. im Schreiber-Modus lange Aufzeichnungen vorgenommen werden und nach einer Störung oder Kalibration der Sensoren der Overload wieder zurück gesetzt werden soll. So kann bei einer **späteren Kontrolle** sofort erkannt werden, ob **wieder ein Überlauf aufgetreten ist** oder die Signale ohne Störungen aufgezeichnet wurden.

Overload ist das **Übersteuern** des dynamischen Bereiches des **ADCs**. Falls dieser z.B. auf 2V, 50% Offset eingestellt ist, liegt der messbare Bereich zwischen -1V und 1V. Falls nun ein Signalpuls von 1.5V auftritt, würde dieser einen Overload auslösen. Dieser wird zur **Laufzeit als roter Balken** oberhalb (bei negativem Overload unterhalb) der Legenden-Kurve dargestellt. Ein **gelber Balken** signalisiert, dass **Vorgängig** ein Über- bzw. Unterlauf **aufgetreten** ist.

Bei jedem "Overload-Zurücksetzen" wird ein **Eintrag im Aufnahmeprotokoll** vorgenommen. Somit sind alle Manipulationen, welche während der Aufnahme vorgenommen wurden protokolliert. Via Menü ***"Ansicht" / "Aufnahmeprotokoll"*** können die eingetragenen Ereignisse dargestellt werden.

Die **Schriftart der Legende** kann man benutzerdefiniert einstellen. So ergibt sich die Möglichkeit einen kleinen Font zu wählen, damit die Legende möglichst schlank bleibt.

14.2 Text-Eintrag in Kurvenanzeige-Fenstern

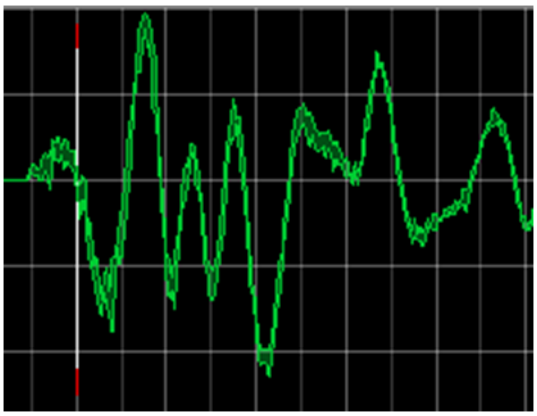
Via "Text" lassen sich Einträge in den Kurvenfenstern vornehmen. Mit Rechts-Klick über einem Text-Eintrag kann danach auch die Schriftart angepasst werden.
Bei der Texteingabe in den Kurvenanzeige-Fenstern können auch Verknüpfungen zu berechneten Einzelresultaten im Formel-Editor eingefügt werden. Dabei müssen die Resultat-Namen als Solche in spitzigen Klammern geschrieben werden. Beispiel: **<VarName>.** Anstelle von *VarName* muss der Resultat-Name geschrieben werden. Falls dieser nicht existiert oder wenn noch keine Berechnung durchgeführt wurde, wird "NotDefined" angezeigt. Vor und nach **<VarName>** kann ein beliebiger Text stehen. Wenn *VarName* ein Zahlen-Resultat ist, kann es sein, dass dieses mit zu vielen Stellen geschrieben wird. Um dies zu vermeiden kann mit der Formel **StringFormat()** die Zahl in einen String gewandelt werden (z.B. xyzStr=StringFormat(xyz,"0.00". Text im Fenster: "... <xyzStr> ...").

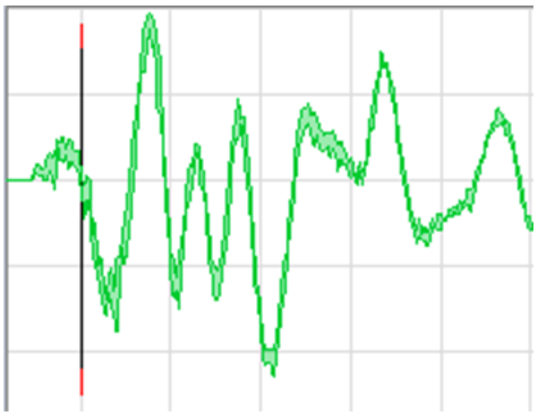
14.3 Gitter

Beim Gitter kann zwischen einem groben und einem feinen Raster gewählt werden. Falls gewünscht kann man dieses auch ganz ausschalten.
Unter dem Menü "Extras/Einstellungen/Benutzeroberfläche..." kann die Farbe des Gitters vorgewählt werden.

14.4 Hintergrundfarbe

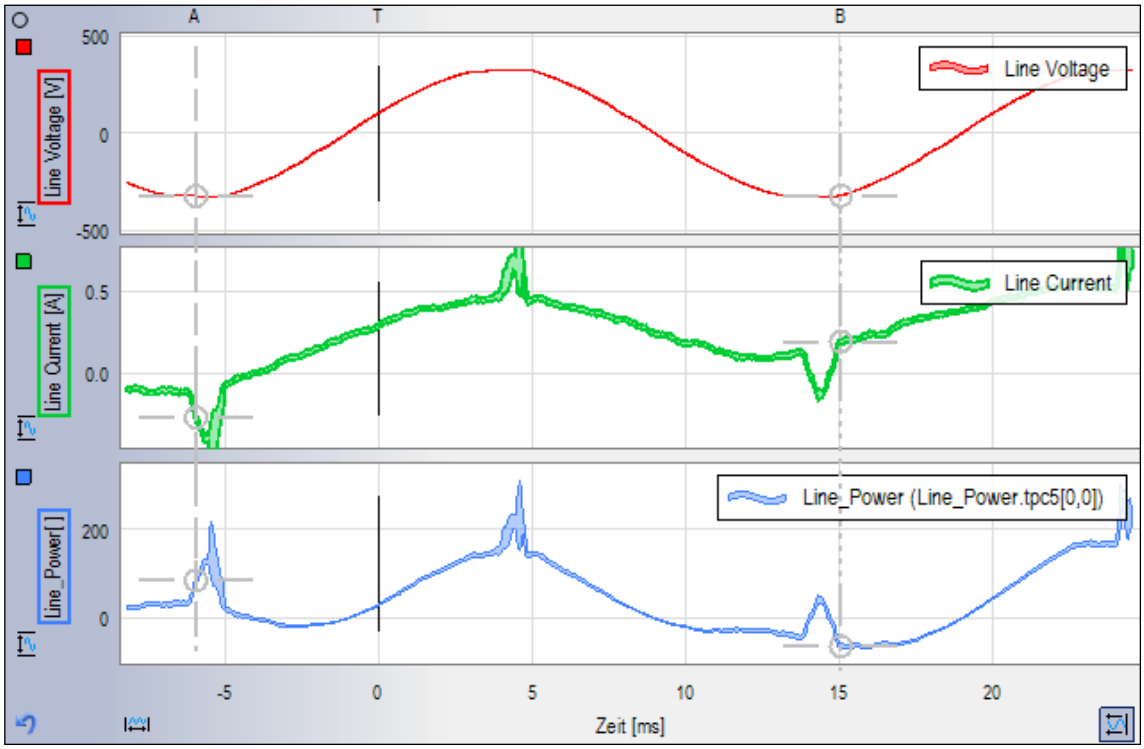
Die Hintergrundfarbe der Anzeige kann man je nach gewünschtem Kontrast entweder auf schwarz oder weiss stellen.
Auch die Hintergrundfarbe kann via Menu "Extras/Einstellungen/Benutzeroberfläche..." vorgewählt werden.





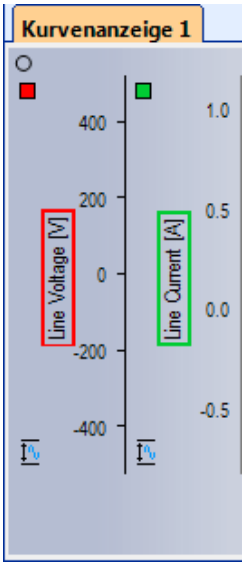
14.5 Bereiche

Anstatt Kurvensignale überlappend anzeigen zu lassen, können diese auch parallel in einzelnen Bereichen dargestellt werden. Es können bis zu 16 Bereiche pro Kurvenanzeige-Fenster eingestellt werden.



14.6 Y-Achsen

14.6.1 Anzahl Y-Achsen links, rechts



Es können für die einzelnen Signale jeweils **separate Y-Skalen** definiert werden. Durch Rechts-Klick in der Kurvenanzeige, "Achsen links" bzw. "Achsen rechts" kann man die Anzahl Skalen bestimmen. Dies kann auch via Menü "Kurvenanzeige" / "Skalen auf der linken Seite" bzw. "Skalen auf der rechten Seite" gemacht werden.

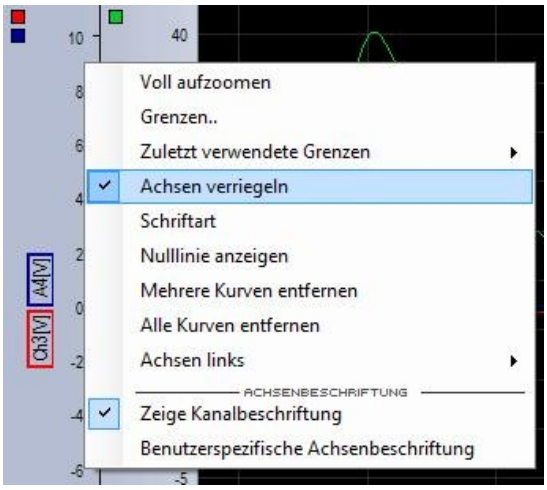
Nachdem zusätzliche Skalen hinzugefügt wurden, können die Kurven mit den farbigen Signal-Kästchen links von der Kurvenanzeige auf die neuen Skalen gezogen werden.
Die einzelnen Y-Skalen lassen sich dann individuell Zoomen und Verschieben.

Es können bis zu 12 Y-Achsen links und rechts erstellt werden.

Man kann der Kurvenanzeige auch vorgängig neue Skalen hinzufügen und anschliessend die Signalkurven direkt vom Control Panel oder Signalquellen-Browser auf die gewünschten Skalen ziehen.

© Elsys AG

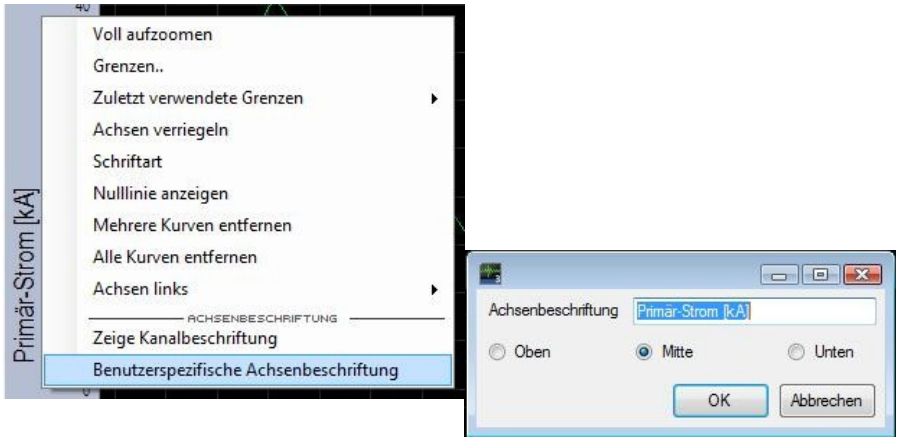
14.6.2 Y-Achsen verriegeln



Wenn in Kurvenanzeige-Fenstern **mehrere Y-Achsen** deklariert wurden, können diese miteinander **verriegelt** werden.
Mit einem Rechts-Klick auf einer der Y-Achsen öffnet man das Pop-up-Menü und wählt *"Achsen verriegeln"*.
Falls man dann jeweils zoomt oder die Kurven verschieben möchte, werden alle Achsen diese Aktion ausführen.

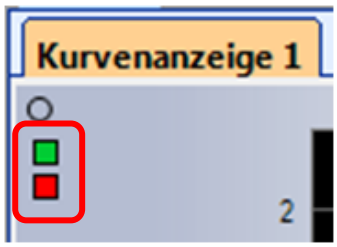
14.6.3 Y-Achsen Beschriftung

Die Beschriftung der Y-Achsen kann auch benutzerspezifisch erfolgen:



Um die angepassten Kurvenanzeige-Einstellungen zu speichern, wählen Sie im *"Datei" / "Layout speichern.."*. In den Layout-Dateien werden die Anordnung von sämtlichen Fenstern und deren inneren Eigenschaften gespeichert.

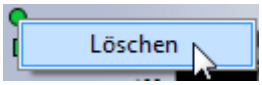
14.7 Kurvendarstellung



Jede Kurve lässt sich in der Kurvenanzeige oben auf den Y-Achsen als kleines **Kästchen** identifizieren. Bei ausgefülltem Kästchen ist die Kurve sichtbar, ein **leeres Kästchen** macht die Kurve **unsichtbar**.
Durch Rechts-Klick auf einem Kästchen kann eine Kurve aus dem Fenster entfernt werden. Zudem können verschiedene Kurveneigenschaften eingestellt werden.
Um die **Reihenfolge** der Kästchen zu ändern, einfach die **Maustaste gedrückt halten** und an einer Position weiter unten loslassen.

Wenn eine Kurve von der **Kurvenanzeige entfernt** wurde, bedeutet dies nicht, dass die Kurvendaten gelöscht wurden! Die Kurven-Eigenschaften (Farbe, Labels, Y-Skalierungen usw.) gehen jedoch verloren.
Um diese Kurve wieder darzustellen, kann sie einfach wieder mittels Drag & Drop vom Control Panel oder vom Signalquellen-Browser in die Kurvenanzeige oder auf eine Y-Achse gezogen werden.

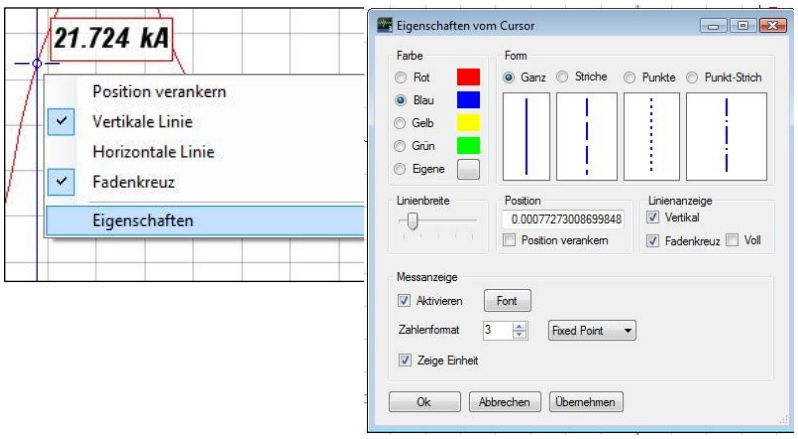
Jede Kurvenanzeige hat einen **Platzhalter** für eine **Statuszeile**, die ein- oder ausgeblendet werden kann mit dem kleinen **runden Knopf** in der linken oberen Ecke des Fensters. Diese Status-Zeile enthält die wichtigsten Informationen von einem beliebigen Signal in der Kurvenanzeige. Mit **Drag & Drop** von einem beliebigen Kurven-Kästchen werden die Informationen dieses Signals in der Statuszeile angezeigt.
Die Statuszeile kann nützlich sein bei der Generierung von Rapporten (via Drucken oder Snapshot), da wichtige Informationen wie Instrument-Name, Kanalname, Zeitstempel, Anzahl Samples und das Abtastinterval festgehalten sind.

	Drag vom Kanal-Schalter …
	… und Drop auf der Statuszeile oberhalb der Kurve.
	Rechtsklick auf den Runden Knopf zeigt die Option um die Statuszeile zu löschen

Falls die Statuszeile nicht mehr erwünscht ist, kann sie entweder mit Links-Klick auf dem runden Knopf ausgeschalten oder per Rechtsklick/Löschen gelöscht werden.

14.8 Cursor-Eigenschaften

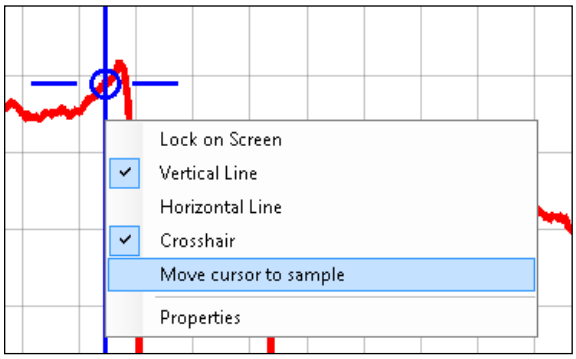
Wenn das Maus-Symbol über einen Cursor platziert wird, kann mit Rechts-Klick die Eigenschaften-Maske geöffnet werden.
Darin können das Aussehen und weitere Eigenschaften des Cursors eingestellt werden
Wenn die "*Messanzeige*" aktiviert ist, werden für jede Kurve die Y-Werte an der Cursor-Position angeschrieben.



Im Menu "Extras -> Einstellungen" hat man in der Rubrik "Benutzeroberfläche" die Möglichkeit, den Cursor Buchstaben (A, B, C …) oben innerhalb der Kurvenanzeige oder auf der Statuslinie darüber darzustellen (die Statuslinie kann via dem kleinen Kreis-Symbol oben links, unsichtbar geschalten werden).

14.9 Cursor auf Abtastpunkten

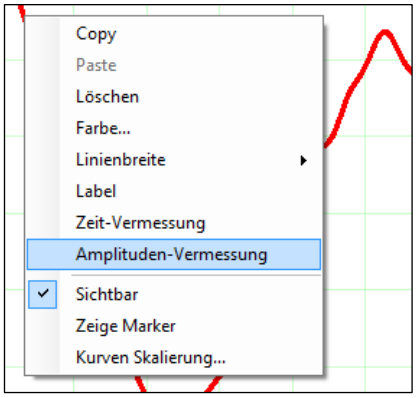
Die Cursors können individuell so eingestellt werden, dass diese nur noch auf den einzelnen Samples bewegt werden können.
Mit einem Rechtsklick auf den Cursor wählt man im Kontextmenu "Bewege Cursor auf Abtastpunkten".



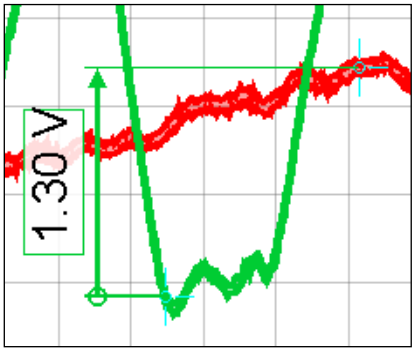
Danach springt der Cursor von Sampling zu Sampling. Zwischenwerte existieren nicht oder allenfalls nur, wenn mehrere Kurven mit unterschiedlichen Abtastraten im gleichen Kurven-Fenster sind.

14.10 Vermessung von Kurven

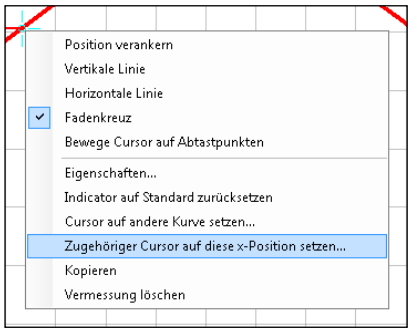
Neben den Labels können auch **Zeit- und Amplituden-Vermessungen** auf die Kurven platziert werden.



Mit einem Rechtsklick auf die entsprechende Kurve erhält man das dazugehörige Kontextmenu.



Hat man die Vermessung platziert, kann man die Cursors in X-Richtung **auseinanderziehen**. Es wird jeweils die Differenz zwischen Cursor 2 und Cursor 1 in einem Textlabel dargestellt. Der Text und das Zahlenformat kann wie gewohnt mit einem Rechtsklick auf den Text via "Text .....bearbeiten.." eingestellt werden.



Um einen der beiden Cursor auf eine **andere Kurve zu setzen**, tätigt man einen Rechtsklick auf den jeweiligen Cursor und wählt im Kontextmenu "Cursor auf andere Kurve setzen..". Danach erscheint ein Dialog, in welchem man das gewünschte Signal auswählen kann.

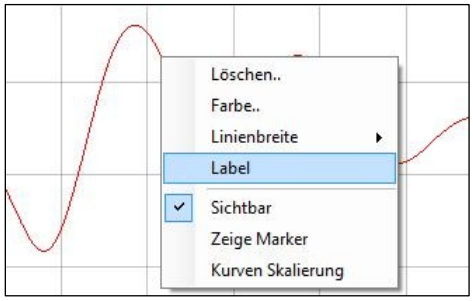
Damit lassen sich Amplitudendifferenzen von zwei Signalen ausmessen (dies macht in der Regel nur Sinn, wenn beide Signale die gleiche Einheit haben).

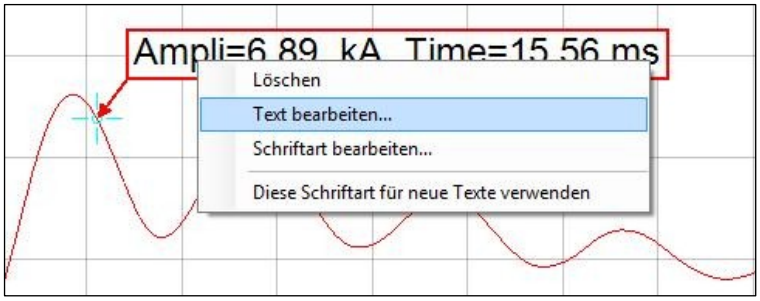
Via Rechtsklick auf "Zugehöriger Cursor auf diese x-Position setzen..." wird der andere Cursor auf die gleiche X-Position gesetzt.

Wenn das Icon ("Cursor miteinander bewegen") gedrückt wird, verschieben sich die Mess-Cursor gemeinsam. Danach lässt sich die Differenz der beiden Kurven an einer beliebigen Stelle auf der X-Achse anzeigen.

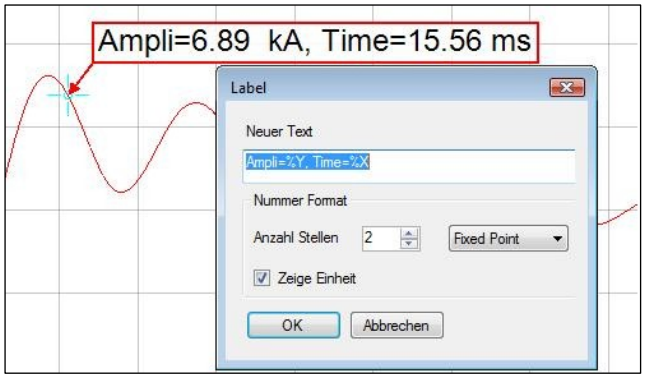
14.11 Labels auf Kurven

Die Kurven können mit sogenannten "Labels" angeschrieben werden. Dazu geht man mit dem Mauszeiger an die Stelle auf der Kurve, an der das Label platziert werden soll. Mit Rechts-Klick und Klick auf "Label" wird die Kurve per default mit dem Amplitudenwert in deren Farbe angeschrieben.





In der Maske, welche sich via Rechts-Klick innerhalb des Labels und Klick auf "Text bearbeiten..." öffnet, kann ein beliebiger Text eingegeben werden.

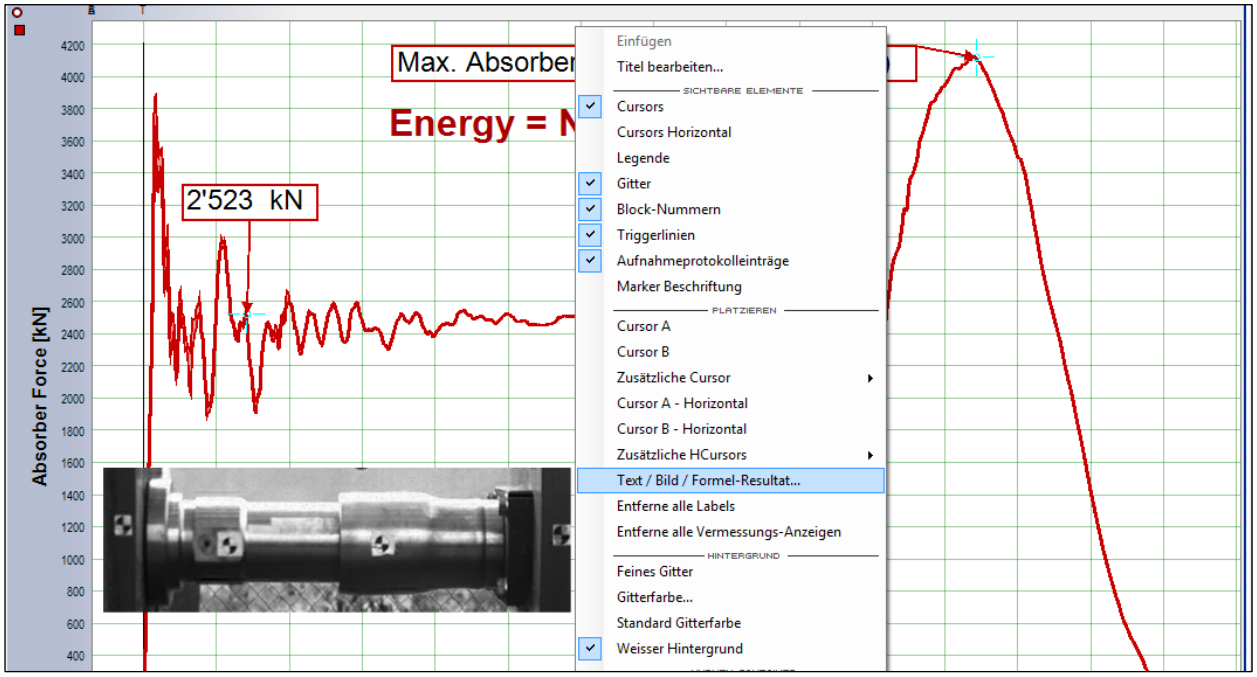


Die Schlüsselbezeichnungen "%Y" bzw. "%X" werden durch die Werte der Y- bzw. Zeit-Achse ersetzt. Zudem können hier die Zahlen-Eigenschaften der Y- bzw. X-Werte definiert werden.

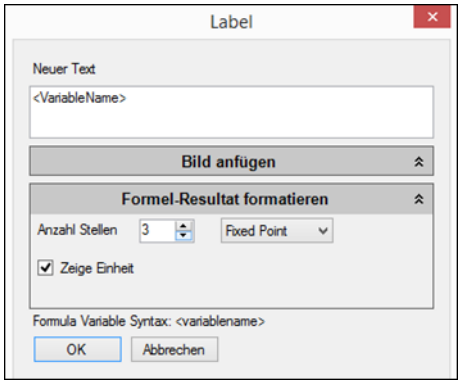
Die Eigenschaften des Label-Fadenkreuzes auf der Kurve können ähnlich wie bei einem Crusor definiert werden. Via dem Fadenkreuz kann das Label auf der Kurve wie ein Cursor verschoben werden.

14.12 Bilder einfügen, Resultate aus Berechnungen formatieren

Via Dropdown-Menu (Rechtsklick in einem Kurvenfenster), kann (wie ein Text) auch ein Bild eingefügt werden.



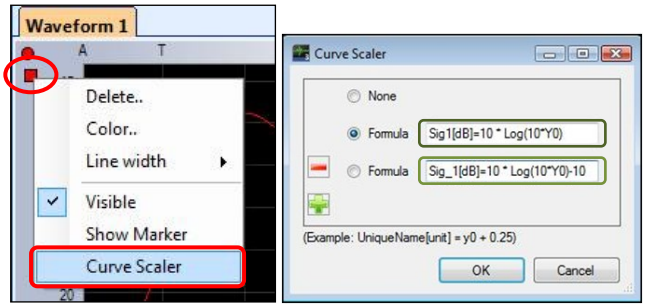
Durch Klicken auf ***Bild anfügen*** kann das gewünschte Bild gesucht und ausgewählt werden. Beim Auswählen wird normalerweise zuerst der Ordner *images* im aktuellen Experiment präsentiert. Durch Navigieren kann jedoch auch ein Bild irgendwo ausgelesen und angefügt werden.



In der gleichen Maske kann auch das Zahlenformat eines, via Formel-Editor berechneten, Resultates festgelegt werden

14.13 Y-Skalierung von Kurven

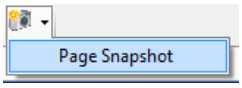
Individuell kann jede Kurve durch eine Formel umgerechnet werden.



Die Formel kann vom Anwender frei **gewählt** werden. Es lassen sich auch vorgängig mehrere Funktionen definieren. Die gewünschte Formel wird dann durch Anschalten des entsprechenden Knopfes aktiviert.
Diese Formeln gehören dann als Eigenschaft zu der Kurve (wie die Farbe). Beim Verschieben bzw. Kopieren (via Drag & Drop) in ein anderes Fenster werden auch die Skalierungs-Formeln mitgeführt.
Es sind natürlich nur Berechnungen gültig, die sich auf Amplituden-Werte (keine X-Achsen-Werte) beziehen. Auch bezieht sich die Umrechnung nur auf die Anzeige der Kurve in einem Kurvenfenster und einer allfälligen Berechnung in einer Skalar-Tabelle. **Speichern von Kurven erfolgt immer unskaliert**.
Bei Funktionen in einer Skalar-Tabelle zu einer derartigen Kurve wird die Y-Skalierung (mit Name und Einheit) auch berücksichtigt.
In den Bereichen, wo die Formel keine Resultate liefern kann (z.B. bei LOG von negativen Werten), wird keine Kurve gezeichnet.
Die Berechnung erfolgt aufgrund der Rohwerte ***y0*** einer Kurve**.** Als Rohwerte gelten auch allenfalls umgerechnete Werte, welche möglicherweise via Formel-Editor entstanden sind oder welche durch entsprechende Einstellungen unter *Physikalische Einheit* im Control-Panel bereits vor der Aufnahme definiert wurden.
Der neue Kanal-Name (ChName[Unit]= ...) darf nicht aus einem Schlüsselwort des Formel-Editors (z.B. Cos, Sin, If, for, to, as, Pi usw.) bestehen.
Fehler in der Skalierungs-Formel werden beim Eingeben mit gekennzeichnet. Bei fehlerhaften Formeln wird die Berechnung unterdrückt. Der Rohwert bleibt stehen.

14.14 Snapshot

Durch Links-Klick über der Schaltfläche wird das Bild der aktuellen Kurvenanzeige in die Windows-Zwischenablage kopiert und kann von dort in eine Benutzer-Anwendung (z.B. in ein Report-Dokument) eingefügt werden.



Wenn mit der Maus auf den Pfeil rechts des Icons geklickt wird, kann der Bildschirm-Inhalt von allen Fenstern in einer Seite (nicht nur von einem Kurven- bzw. Skalar-Fenster) in die Zwischenablage kopiert werden.

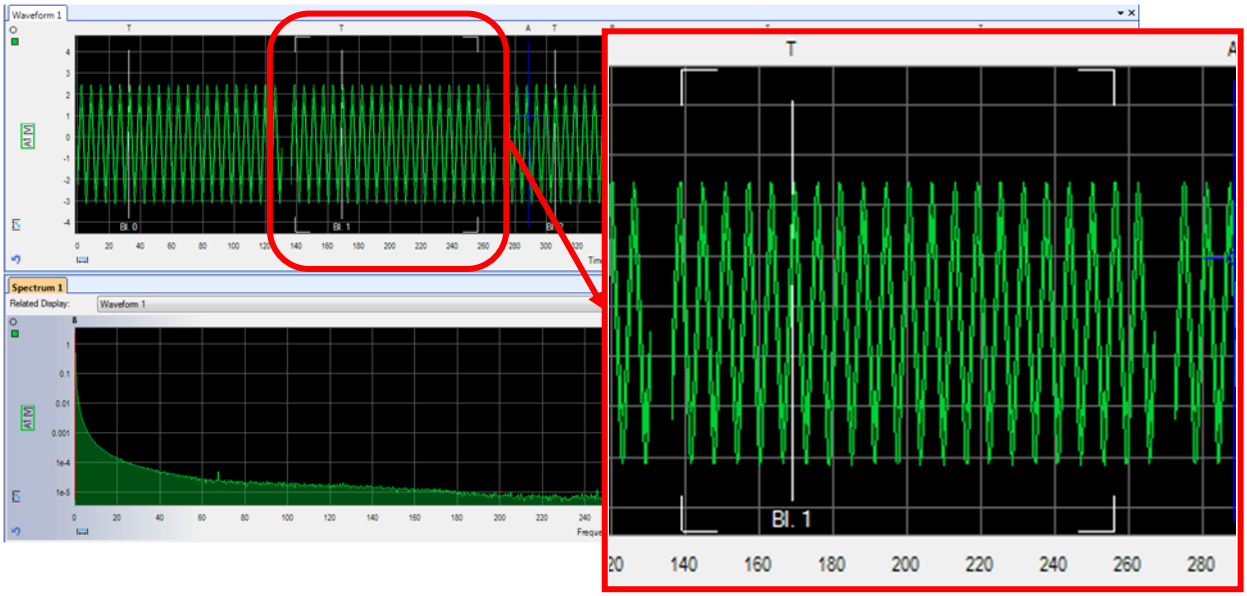
Unter "Extras / Einstellungen / Benutzeroberfläche / Snapshot" kann zwischen "Bitmap" und "Vektor" gewählt werden.
Zudem kann die Grösse, wie der Bildausschnitt danach erscheinen soll, gewählt werden. Bei mehreren Fenstern in der Seite werden diese proportional angepasst. Alle Text-Objekte werden unverzerrt übertragen. Dies sollte beachte werden, wenn in einer Kurvenanzeige benutzerspezifischer Text eingetragen wurde. Dieser kann dann anders platziert oder gar abgeschnitten erscheinen.
Hier kann auch eingestellt werden, dass das Bild immer mit weissem Hintergrund in der Zwischenablage abgelegt wird. Falls dabei auf dem Bildschirm mit schwarzem Hintergrund gearbeitet wird, sollten die Kurven-Farben eher dunkel gewählt werden.

14.15 Analyse von Multiblock-Aufnahmen (Block-Jumping)

Möchte man nach einer Multiblock- oder ECR-Aufnahme von Block zu Block springen, kann man dazu über die Toolbar folgende Steuerelemente verwenden:

Man erteilt dem gewünschten Fenster den Fokus und klickt auf das jeweilige Steuerelement. Anstatt über die Toolbar die Steuerelemente zu bedienen, kann man auch ganz bequem mit den Tasten "Page UP", "Page DOWN" arbeiten. Mit den Tasten "Home“ und "End“ springt man jeweils auf den ersten bzw. den letzten Block.

Falls ein Zweitfenster (XY, Marker, FFT, Zoom) den Fokus hat und dann von Block zu Block gesprungen wird, bewegen sich im Hauptfenster nur die entsprechenden Zeit-Marken



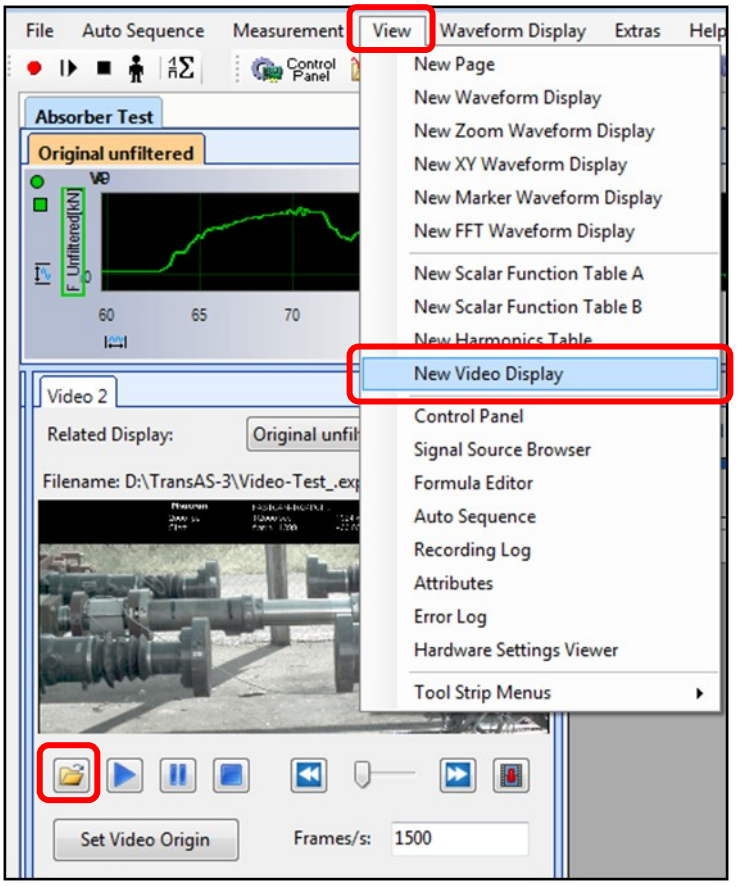
15 Videos anzeigen, synchron zu Signal-Kurven

Video-Filme können in TranAX vor-und rückwärts abgespielt oder einfach als Standbild angezeigt werden. Filme lassen sich parallel zu aufgezeichneten Signal-Verläufen abspielen. Wenn nun diese Signale gleichzeitig mit den Videos aufgenommen wurden, lässt sich die Bildfolge synchron zu den Kurven-Verläufen analysieren.

Die Filme können in gängigen Formaten (avi, mp4, mpg usw.) aufgenommen worden sein. Am besten eignet sich das avi-Format. Vorbereitend sollten die Video-Dateien in ein Unterverzeichnis ***"videos"*** des TranAX-Experiments mit den zugehörigen Kurven-Dateien kopiert werden.

Datei Typ	**Verzeichnis**
Kurven-Dateien (*.tpc5)	*...\experimentName.exp**data** Standard Verzeichnis der Kurven-Dateien
Filme (*.avi, *.mpg, usw.)	*...\experimentName.exp**videos** Ev. dieses Verzeichnis vorab manuell erstellen

Für die Darstellung von Filmen benötigt TranAX ein zusätzliches Programm namens VLC Media Player (vlc.exe). Falls dieses Programm nicht bereits vorhanden ist, kann es von der Seite http://www.videolan.org heruntergeladen und installiert werden.

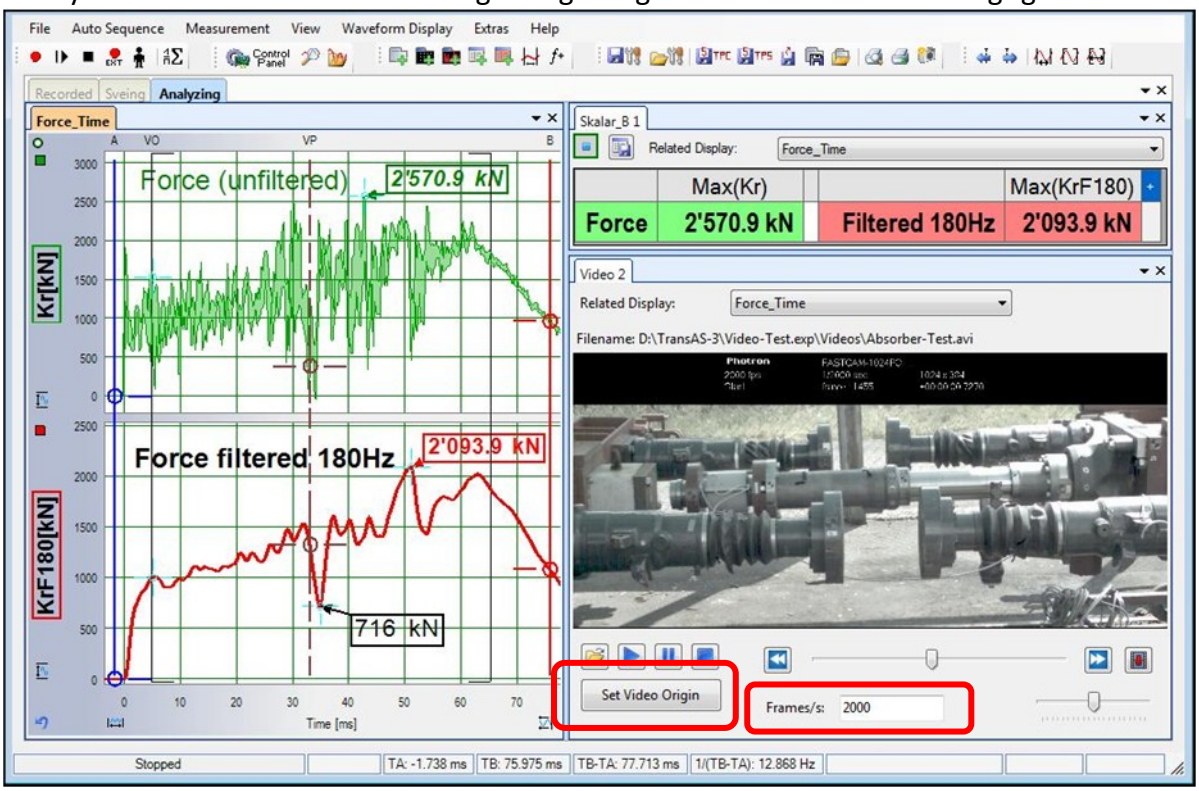


Ein Video-Fenster wird via Menu "Ansicht" geöffnet und kann danach, falls erforderlich, wie gewohnt umplatziert werden.
Beim Öffnen eines Video-Fensters wird auch gleich der Dialog zur Auswahl einer Video-Datei geöffnet.

Der Parameter "Zugehöriges Fenster" wird per Standard auf das aktivierte Hauptfenster mit Signal-Kurven gesetzt. Dieser kann auch im Nachhinein angepasst werden.

Via kann auch ein anderer Video-Film geladen werden

Die Synchronisation des Films mit zugehörigen Signal-Verläufen wird wie folgt gemacht:



- Zugehörige **Kurven-Signale** in das entsprechende Anzeigefenster **platzieren** (z.B. via Signalquellen-Browser). Ev. Zoomen und Verschieben der Kurve auf eine **markante Stelle im Signalverlauf**, wo auch eindeutig das entsprechende Bild im Film zugeordnet werden kann.
- An dieser markanten Stelle via **rechte Maus-Taste** in der Kurvenanzeige den **"Video Origin Marker"** setzen.
- Mit dem Parameter **"Frames/s" die Bildfrequenz** des Videos einstellen (dieser Wert kann nicht aus allen Video-Formaten übernommen werden, er sollte also in jedem Fall überprüft und richtig eingestellt werden).
- Den Film-starten und an der **markanten Stelle stoppen**.
- Danach mit Hilfe des Knopfes ***"Set Video Origin"*** das aktuelle **Bild** mit dem **Origin auf der Kurve synchronisieren**. Dabei wird der "Video Play Marker" auf das Video-Origin der Signal-Kurve gesetzt. Durch Verschieben des Video-Play-Markers kann das gewünschte Bild noch feiner angefahren werden. Ev. durch Aufzoomen an der entsprechenden Stelle den Kurven-Ausschnitt vergrössern. Danach erneut "Set Video Origin" anklicken.

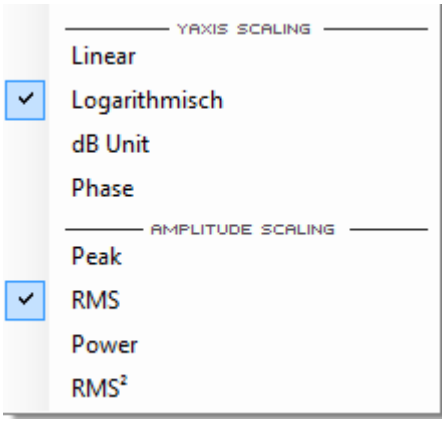
Der Film kann nun vor- und rückwärts gespult werden. Dabei läuft der Video-Play-Marker im Kurven-Fenster mit. Wenn umgekehrt der Video-Play-Marker verstellt wird, läuft auch der Film entsprechend mit. Die Synchronisation sollte also gewährleistet sein.

Die korrekte Übereinstimmung ist natürlich nur gewährleistet, wenn bei ***"Frames/s"*** die richtige Bildfrequenz eingestellt ist.

© Elsys AG

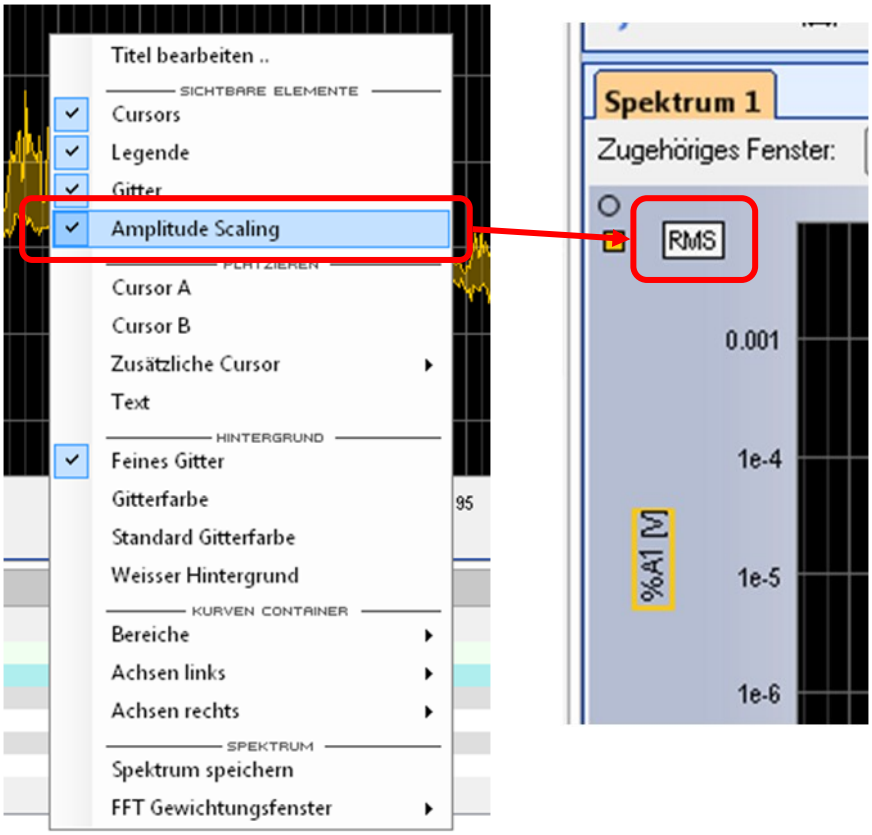
16 FFT Kurvenanzeige

16.1 Darstellung Skalierung



Bei einer FFT-Kurvenanzeige kann die Skalierung einer Y-Achse gewählt werden. Ein Rechtsklick auf der gewünschten Skala öffnet das entsprechende Popup-Menü.
Entweder wird die Achse **Linear**, **Logarithmisch oder als dB Unit** dargestellt. Bei "Phase" erfolgt das Anzeigen des Phasenverlaufs.
Die Skalierung kann als **Peak**, **RMS**, **Power** oder **RMS²** gewählt werden.

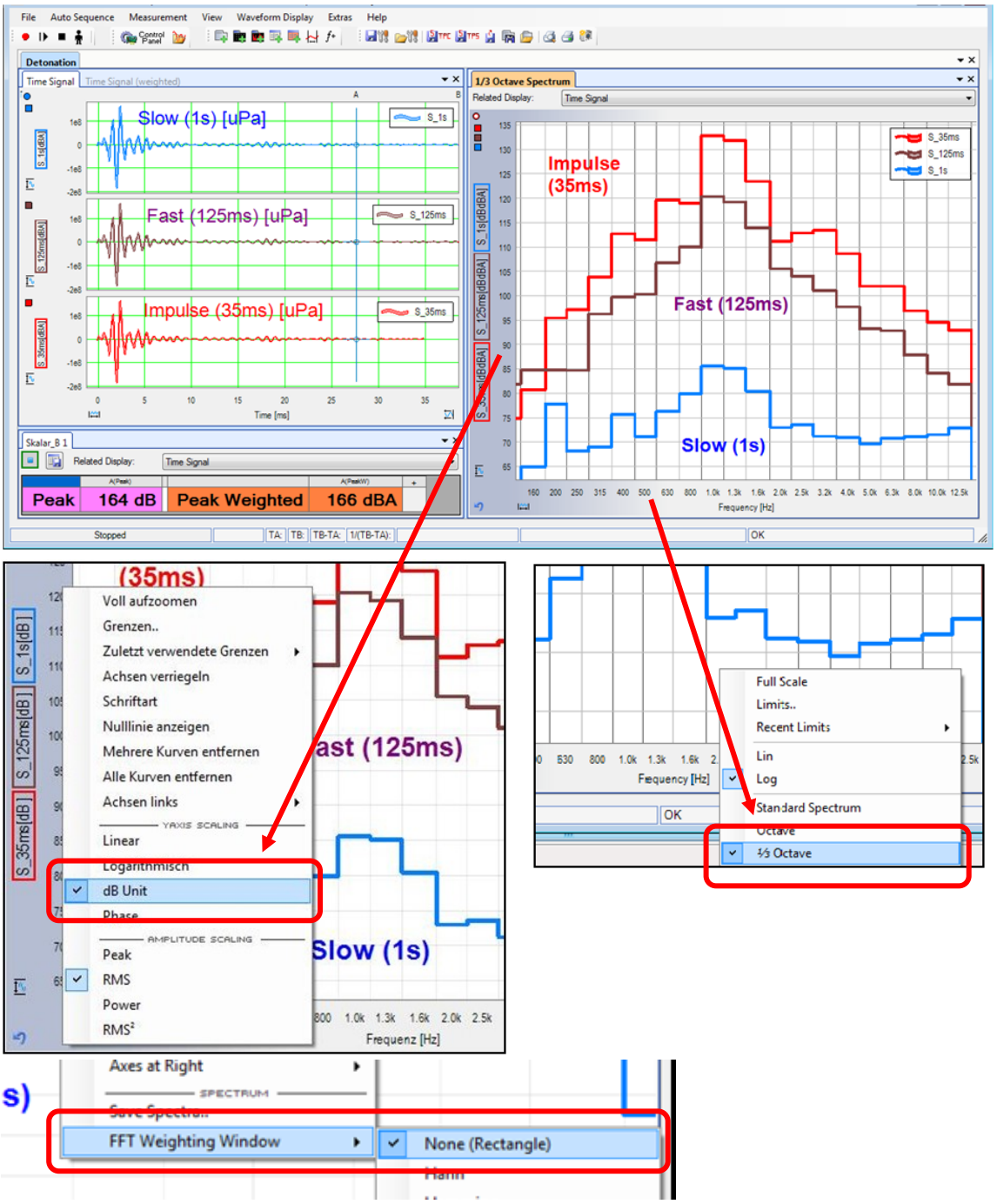
Um die eingestellte Skalierung oben auf den Y-Achsen anzuzeigen, kann durch einen Rechtsklick in die Kurvenanzeige die Option **"Amplitude Scaling"** eingeschalten werden.



16.2 Oktaven- und Terz-Anzeigen

Das FFT Waveform Display wurde neben der bestehenden Darstellung als "Standard Spektrum" um die Optionen "Oktaven" und "⅓ Oktaven" (Terz) erweitert. Diese dienen vor allem für die Analyse bei akustischen Anwendungen.

Mit einem Rechtsklick auf der **Frequenz-Skala** kann im Pop-Up-Menu die "Octave" oder "1/3 Octave" (Terz) Darstellung gewählt werden. Für Octave- oder Terz-Darstellung muss die **Einheit auf der Y-Achse auf "dB Unit"** gestellt sein.

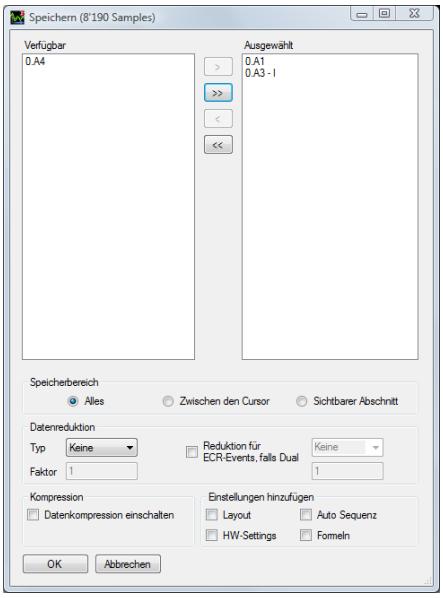


Normalerweise muss für Octave- oder Terz-Darstellung das **FFT-Gewichtungs-Fenster** ausgeschaltet werden (**"Rechteck"**).

17 Aufnahmen Speichern & Drucken

17.1 Speichern

Mit Hilfe der Schaltfläche oder Menü ***"Datei" / "Speichern als tpc5-Datei"*** können Signalkurven im HDF5-Format auf die Festplatte gespeichert werden. (Das HDF5-Format wurde definiert von NCSA, dem "National Center for Supercomputing Applications", USA).

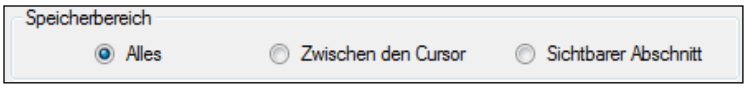


Die linke Liste zeigt die **verfügbaren Signale**, die zum Abspeichern ausgewählt werden können. Die **rechte Liste** zeigt die aktuell ausgewählten Kurven.
Zur Auswahl stehen alle Hardware-Kanäle des aktuellen Clusters sowie alle Kurven aus Dateien, welche sich in einem Kurvenanzeige-Fenster befinden.

Weitere Optionen zur Komprimierung finden Sie im Menü ***"Extras" / "Einstellungen" / "Import/Export"***.

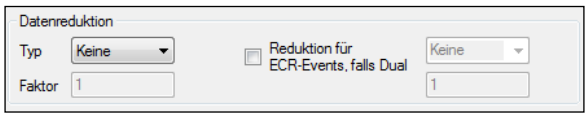
Wenn die gespeicherten Kurven später wieder in TranAX angezeigt werden sollen, müssen sie auf diese Weise im HDF5-Format (*.TPC5) abgespeichert werden.

17.1.1 Speicherbereich



Es kann ausgewählt werden, ob man die **gesamte Aufnahme**, nur den Bereich **zwischen den Cursor (A und B)** oder den aktuell **sichtbaren Bereich** abspeichern möchte.

17.1.2 Datenreduktion

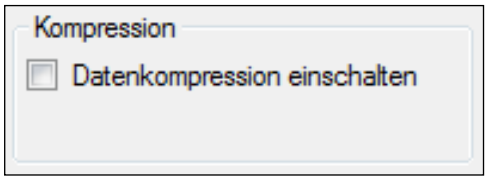


Die **Datenmenge** kann beim Abspeichern **verringert** werden. Es stehen verschiedene Varianten zur Verfügung:

- **Keine**: Keine Datenreduktion
- **Überspringen**: Es wird nur jedes n-te Sample abgespeichert.
- **Mittelung**: Für je n Samples wird der **Mittelwert** berechnet und dieser Wert wird abgespeichert.
- **MinMax**: Für je n Samples wird die kleinste und die grösste Amplitude gesucht und diese Werte werden abgespeichert. Bei dieser Variante werden für jede ausgewählte Kurve zwei Kurven abgespeichert, eine für die untere und eine für die obere Hüllkurvenbegrenzung.

Falls die Datenaufnahme im **ECR-Mode mit Dual** durchgeführt wurde, so kann für die schnellen ECR-Ereignisse und den Dual-Anteil eine unterschiedliche Datenreduktion gewählt werden.

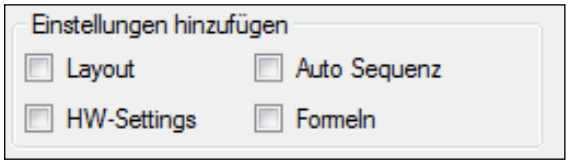
17.1.3 Kompression



Zusätzlich zur Datenreduktion können die Daten **komprimiert** werden. Das führt zu **10% bis 15%** kleineren Dateien, je nach Kurveninhalt.

Die **Kompression** ist **verlustfrei**, es gehen keine Daten verloren!
Je nach Menge der Daten kann das Speichern mehr Zeit in Anspruch nehmen.

17.1.4 Einstellungen hinzufügen

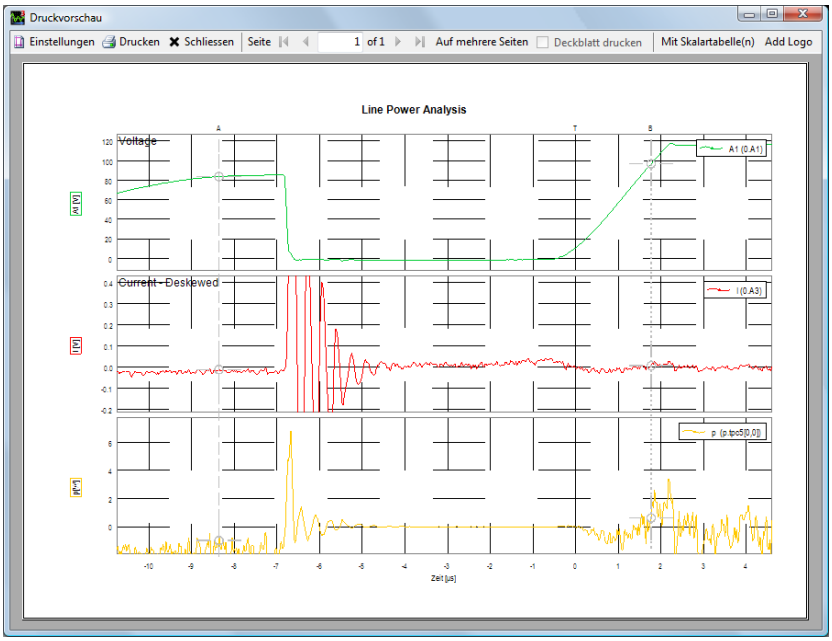


Das TPC5-Format erlaubt nicht nur das Speichern von Messkurven sondern auch dasjenige des **Layouts**, der **HW-Einstellungen**, der **Autosequenz** und des **Formeln-Editors**.

17.2 Exportieren

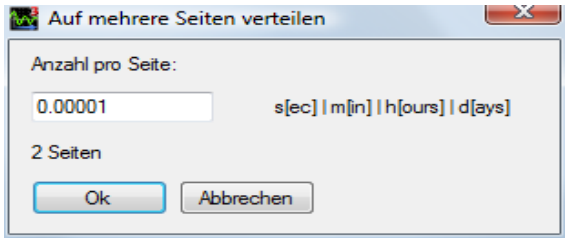
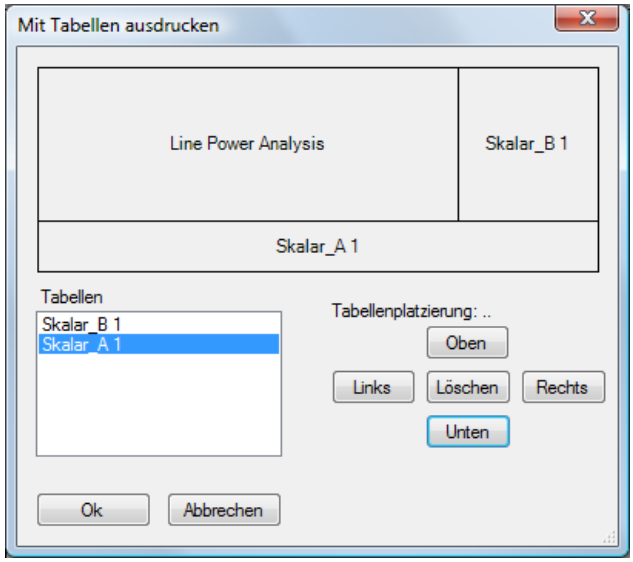
Mit der Schaltfläche oder im Menü ***"Datei" / "Kurven Exportieren"*** können Signalkurven in den Formaten TPC (TransAS 2 - kompatibel), ASCII, DIAdem, Krenz usw. exportiert werden.

17.3 Drucken



Um die aufgenommenen Kurven zu drucken wird zuerst das gewünschte **Kurvenfenster selektiert** und anschliessend die Schaltfläche Drucken gedrückt. Alternativ kann im Menü *"Datei / Drucken."* gewählt werden.
Die Druckvorschau und die Druck-Einstellungen finden sich unter Menü "*Datei / Druckvorschau*". Es kann auch die Schaltfläche geklickt werden um zur Druckvorschau zu gelangen.

© Elsys AG

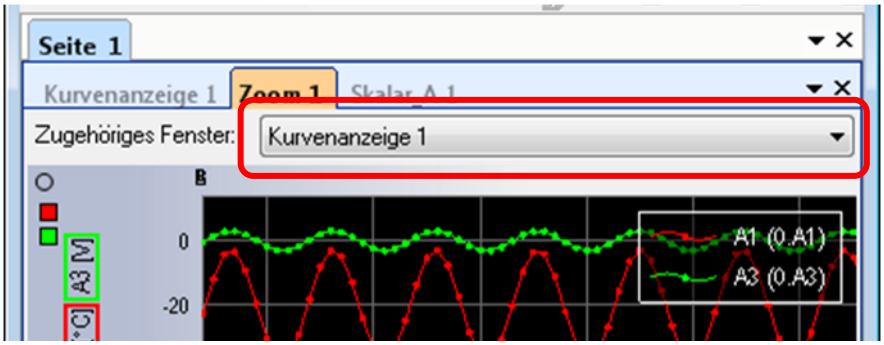
Einstellungen	Hier werden der Drucker, die **Papiergrösse** und die Orientierung gewählt.
Drucken	Dieser Knopf muss gedrückt werden, um die aktuelle **Kurvenanzeige** mit den allenfalls zugehörigen **Skalartabellen auszudrucken**.
Auf mehrere Seiten	Die **Zeitachse** kann zum Ausdrucken über **mehrere Seiten** gedehnt werden. Nachdem die Zeitspanne pro Seite eingegeben wurde, steht unterhalb der Eingabe auf wie viele Seiten der Ausdruck zu stehen kommt. 
Deckblatt drucken	Falls Attribute oder Fehlerprotokolle vorhanden sind, können diese Einträge an dieser Stelle in Tabellenform gedruckt werden.
Mit Skalartabelle(n)	Falls ein Kurvenfenster zugehörige Skalartabellen hat, muss definiert werden, an welcher Position diese Tabellen relativ zum Kurvenfenster ausgedruckt werden sollen. 
Add Logo	Dieser Knopf befindet sich oben rechts im Fenster Druckvorschau. Sollte ein Logo mit auf den Ausdruck, so wird das ausgehend von Add Logo hinzugefügt. Unterstützt werden die Formate bmp, jpg, gif und png.

17.4 Zusätzliche Kurvenanzeigen

Neben der regulären YT - Kurvenanzeige gibt es noch **zusätzliche Arten** von Anzeige-Fenstern. Sie werden über das Menü ***"Ansicht"*** aufgerufen:

- Neue **Zoom-Kurvenanzeige**: Um einen Bereich vergrössert darzustellen.
- Neue **XY-Kurvenanzeige**: Zeigt eine XY-Darstellung an.
- Neue **Marker-Kurvenanzeige**: Zeigt die digitalen Marker Signale.
- Neue **FFT-Kurvenanzeige**: Um das Frequenz-Spektrum des Signals anzuzeigen.

Durch Hinzufügen eines dieser Kurvenanzeige Fenster, können **Signalkurven** gleichzeitig mit **unterschiedlichen** Einstellungen **dargestellt** werden.
Eventuell muss das zugehörige Kurvenanzeige-Fenster bestimmt werden (siehe Bild).



Die zugehörige **Kurvenanzeige** dient hauptsächlich zur Festlegung von **Begrenzungs-Marken** für den Zeitausschnitt, auf welche sich die Signale in den speziellen Anzeige-Fenstern beziehen.
Eventuell müssen zusätzlich gewünschte Kurven via **Drag & Drop** aus dem Control Panel oder dem Signalquellen-Browser geholt werden. Die Kurven können aber auch direkt via Drag & Drop aus dem **ursprünglichen Fenster** kopiert werden.
In der zugehörigen Kurvenanzeige erscheinen nun **vier weisse Ecken (Zeitfensterbegrenzungen)**, die den Bereich aufspannen, der für das hinzugefügte Kurvenfenster relevant ist. Diese Markierungen können auch mit der **Maus verschoben** werden.
Für eine **Zoom-Kurvenanzeige** heisst das, dass der (im zugehörigen Fenster) markierte Bereich **vergrössert** dargestellt wird.
Bei einer **FFT-Kurvenanzeige** wird das **Spektrum aus dem markierten Bereichs** berechnet und angezeigt. Analoges gilt für die XY- und Marker-Darstellung.

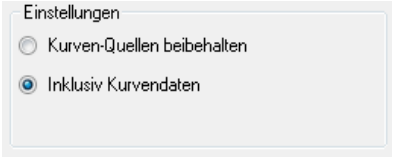
Die zusätzliche Kurvenanzeige-Fenster können auch (wie alle Fenster und Tabellen) an beliebigen Seiten an den andern Fenstern angedockt werden (Siehe dazu auch das Kapitel TranAX Übersicht).

17.5 Speichern von Kurvenanzeige-Seiten

Mit der Aktion **"Seite speichern"** kann eine Untermenge von **allen Layout-Einstellungen** inklusiv Messdaten verwaltet werden. Dabei werden alle Anzeige-Fenster und Skalar-Tabellen in der entsprechenden Seite (mit dem Fokus) in eine Datei gespeichert.
Falls in der Maske dazu unter Einstellungen **"Inklusive Kurvendaten"** gewählt ist, werden von allen in den Anzeigefenstern enthaltenen Kurven auch deren Kurvendaten mitgespeichert. Dabei werden auch die Quellen-Angaben der Kurven in den Fenstern angepasst, sodass nach dem Laden einer gespeicherten Seite auch auf die Kurven in der Datei gezeigt wird. Dieses Feature ist vor allem nützlich, wenn bei einer **späteren Auswertung** das **Erscheinungsbild identisch** sein soll wie bei der Aufnahme, wo die Kurvendaten in der Regel unmittelbar von den Hardware-Kanälen her kommen. Bei der Auswertung wären die Kurvendaten ja nicht mehr unter diesen Quellenangaben vorhanden.

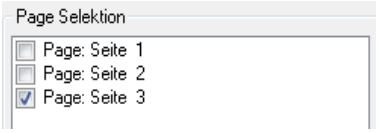
Das Speichern von Seiten kann über das Menü ***"Datei" / "Seite speichern"*** oder mit dem entsprechenden Icon auf der Toolbar eingeleitet werden.

Danach öffnet sich der Dialog, um die Seite(n) zu speichern. Nun gibt man den gewünschten Dateinamen ein. Auf der rechten Seite hat man globale Einstellungsmöglichkeiten:



Kurven-Quellen beibehalten:
Es werden lediglich die **Referenzen** (Quellen-Angaben) auf die Kurven in die Seiten-Datei gespeichert.

Inklusiv Kurvendaten:
Es werden auch **alle Kurven** in die Datei abgespeichert und die Referenzen entsprechend angepasst.



Hier kann man auswählen, welche Seiten in die Datei gespeichert werden sollen. Beim Öffnen des Dialogs ist jeweils nur die Seite mit dem Fokus angewählt.

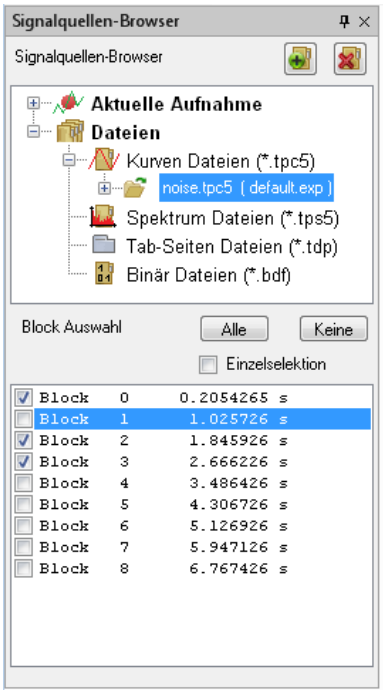
Die Datei wird mit der Endung ***.tdp** in das Data Verzeichnis des aktuellen Experiments abgelegt.

Über das Menü *"Datei" / "Seite laden"* oder das entsprechende Steuerelement auf der Toolbar können gespeicherte Seiten wieder geladen werden. Im Dialog wählt man die entsprechende Datei und die daraus zu ladenden Seiten.

Die Seiten-Datei (*.tdp) kann ebenfalls im Signalquellen-Browser geöffnet werden. Daraus kann dann auf die gespeicherten Kurven zugegriffen werden.

18 Signalquellen-Browser

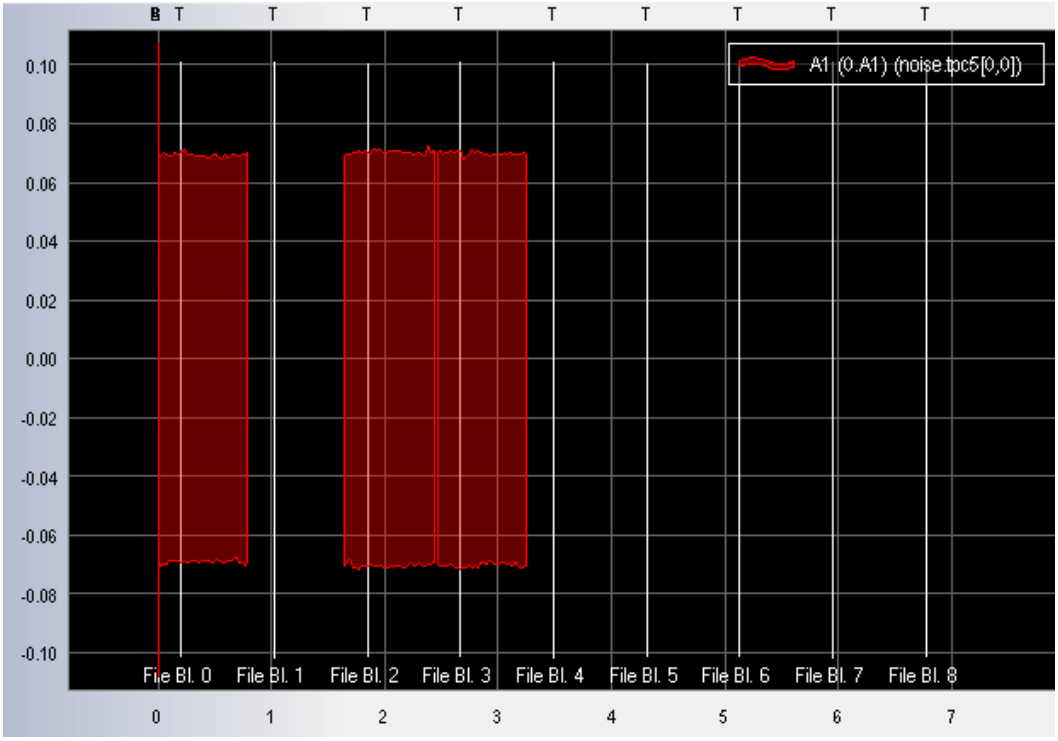
Der Signalquellen-Browser kann im Menü *Ansicht* oder mit der Schaltfläche geöffnet werden. Er wird in der Regel am linken Bildrand platziert.



Zusätzlich zum Control Panel werden auch alle Kanäle im **Signalquellen-Browser** unter "Aktuelle Aufnahme" aufgelistet. Mittels Drag & Drop werden die Daten als Signalkurven in der Kurvenanzeige dargestellt.
Um auf gespeicherte Kurven zugreifen zu können, kann via die Schaltfläche (oben rechts im Browser-Fenster) auf die gewünschte TPC5-, TPS5- oder TPD -Datei navigiert werden.

Für Aufnahmen im Multiblock- oder ECR-Mode zeigt der untere Bereich des Fensters jeden aufgezeichneten Block zum entsprechenden ausgewählten Kanal. Mit der Schaltfläche **"Keine"** werden alle Blöcke deaktiviert, die **"Alle"**-Schaltfläche zeigt wieder alle Blöcke an.

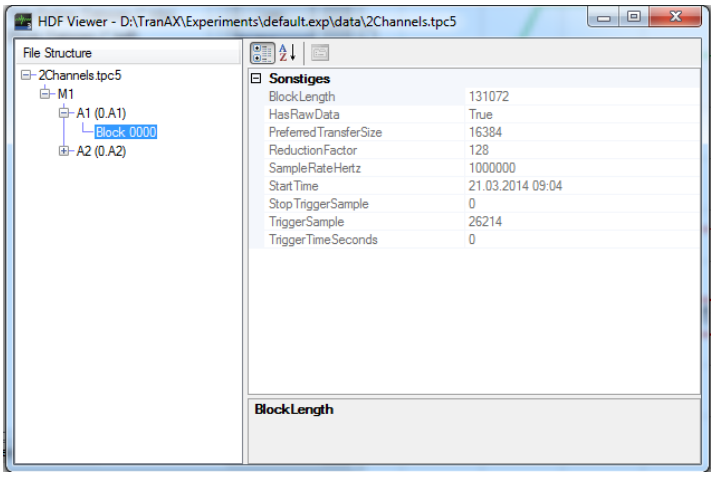
Mit einem Klick auf die entsprechende Schaltfläche werden Blöcke entweder hinzugefügt oder entfernt. Die ausgewählten Blöcke mit dem Häklein werden dann in der Kurvenanzeige dargestellt. Hier im Beispiel werden die drei Blöcke 0, 2 und 3 dargestellt.



 © Elsys AG

18.1 HDF Viewer

Mit dem HDF Viewer kann man eine HDF-Datei (*.TPC5, *.TPS5, *.TDP) betrachten. Man erhält alle dazugehörigen Metainformationen, welche in die Datei gespeichert wurden. Mit einem Rechtsklick auf eine HDF-Datei wählt man im Kontextmenu "HDF Viewer öffnen".



18.2 Excel-Importer

Wenn man beim Setup den **Excel-Importer** installiert hat, erhält man im Kontextmenu einer HDF5 Datei den Eintrag "Nach Excel exportieren...". Somit hat man die Möglichkeit direkt aus dem Signalquellen Browser eine oder mehrere Kurven in eine Excel-Mappe zu übertragen.

19 Skalarfunktionen

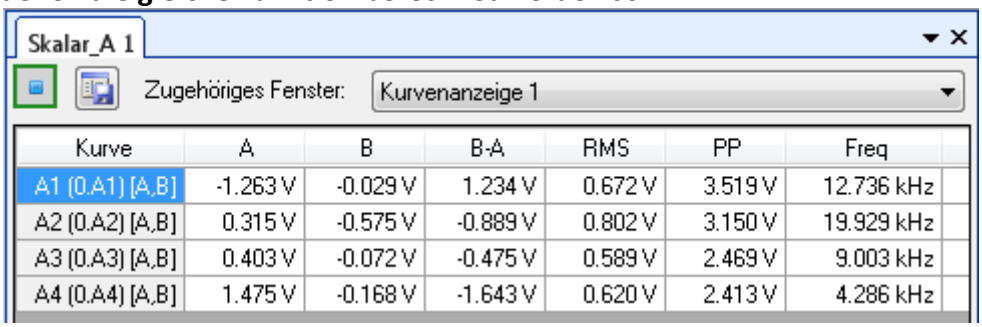
TranAX gibt direkten Zugriff auf **über 60 eingebaute Skalarfunktionen**. Eine Skalarfunktion ist eine Messung oder **Berechnung** von **Kurven-Parametern** wie Maximum, Spitzenwert, Effektivwert oder der Phase zwischen zwei Kurven.

Eine komplette Liste der Skalarfunktionen ist im Anhang gelistet.

Auf den Kurven können Markierungen angezeigt werden, die sich auf entsprechende Resultate bei Skalar-Berechnungen beziehen. Es sind dies kleine Kreise bei Amplituden-Werten (z.B. Maximum, Peak-Peak usw.) oder kleine Rechtecke bei Zeit-Angaben (z.B. RMS-Periodic usw.). Dies erfordert eine globale Einstellung unter "Extras / Einstellungen / Benutzeroberfläche / Markierungen auf Kurve anzeigen".

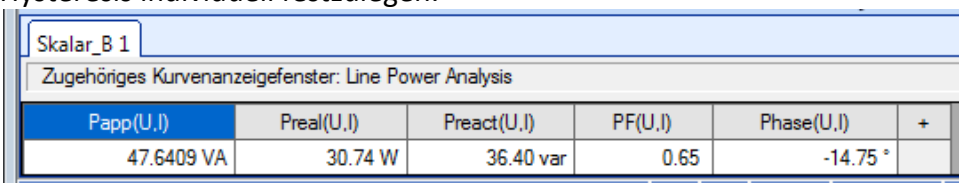
Die Skalarfunktionen können in zwei verschiedenen Tabellen aufgesetzt und konfiguriert werden. Das erlaubt das Ausmessen von mehreren Kurven aus der letzten Datenerfassung oder abgespeicherten Kurven aus Dateien.

Die Skalarfunktionen Tabelle A ist eine matrixförmige Struktur von Kurven gegenüber Skalarfunktionen. Sie ist sehr schnell aufgebaut und ideal für Anwendungen mit **vielen Kurven,** aus denen **die gleiche Funktion** berechnet werden soll.



Kurve	A	B	B-A	RMS	PP	Freq
A1 (0.A1) [A,B]	-1.263 V	-0.029 V	1.234 V	0.672 V	3.519 V	12.736 kHz
A2 (0.A2) [A,B]	0.315 V	-0.575 V	-0.889 V	0.802 V	3.150 V	19.929 kHz
A3 (0.A3) [A,B]	0.403 V	-0.072 V	-0.475 V	0.589 V	2.469 V	9.003 kHz
A4 (0.A4) [A,B]	1.475 V	-0.168 V	-1.643 V	0.620 V	2.413 V	4.286 kHz

Die Skalarfunktionen Tabelle B auf der anderen Seite ist **sehr flexibel** und wurde entwickelt für **komplexe Skalarfunktionen**. Diese Art von Tabelle gibt die Möglichkeit, per Funktion eine Primärkurve, falls benötigt eine Referenzkurve, das Cursor-Paar sowie die Messschwelle mit Hysteresis individuell festzulegen.



Papp(U,I)	Preal(U,I)	Preact(U,I)	PF(U,I)	Phase(U,I)	+
47.6409 VA	30.74 W	36.40 var	0.65	-14.75 °	

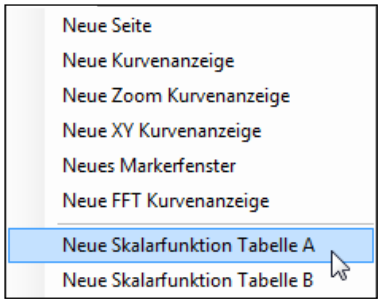
Die Kapitel Skalarfunktionen Tabelle A und Skalarfunktionen Tabelle B beschreiben Schritt für Schritt, wie eine neue Skalarfunktion Tabelle mit den gewünschten Kurven geöffnet und konfiguriert wird.
Die Skalarfunktionen sind im TranAX Skalarfunktions-Dialogfenster und im Kapitel Beschreibung der Skalarfunktionen dieses Handbuches dokumentiert.

Eine neue Skalarfunktions-Tabelle wird entsprechend der Grundeinstellung platziert, die im Menü ***"Extras" / "Einstellungen" / "Benutzeroberfläche" / "Standard Tab-Fenster-Platzierung"*** definiert ist.

Die Werte der markierten Skalartabellen können mit **copy & paste** (<Ctrl>+c / <Ctrl>+v) in eine andere Anwendung (z.B. Excel) übertragen werden.

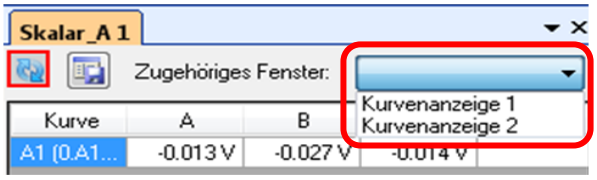
Mit **<Shift> + <Ctrl>+c** wird bei der **Skalartabelle A** auch die **Titellinie** mit übertragen.

19.1 Skalarfunktionen Tabelle A



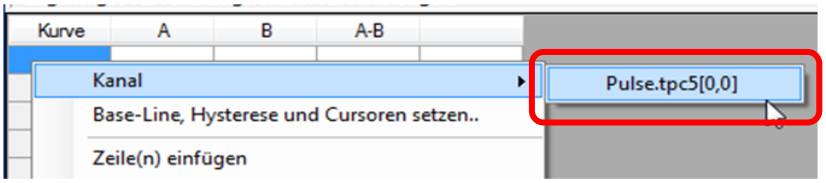
Eine neue Skalarfunktions-Tabelle kann mit Menü ***"Ansicht" / "Neue Skalarfunktion Tabelle A"*** geöffnet werden. Innerhalb des Kurvenfensters wird ein neues Registerblatt mit der Skalarfunktion-Tabelle kreiert. Alternativ kann auch das Icon im Menü geklickt werden.

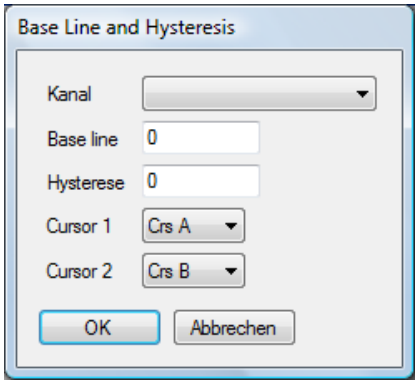
Standardmässig werden die **Amplitudenwerte von Cursor A, Cursor B** und deren **Differenz** in die Tabelle eingefügt. Das zugehörige Kurvenfenster, also die **Quelle** der zur Verfügung stehenden Kurven, kann im Dialog **"Zugehöriges Fenster"** ausgewählt werden. Hier werden alle verfügbaren Kurvenanzeigen aufgelistet. Die neu erstellte Tabelle kann nun sehr **flexibel positioniert** werden. Durch Ziehen des Registerblattes ***"Skalar_A 1"*** lässt sich die Tabelle auf einer beliebigen Seite des aktiven Kurvenfensters platzieren. Natürlich kann die Tabelle auch als eigenes Registerblatt verwendet werden.



19.1.1 Kanalauswahl zur Berechnung der Skalarfunktionen

Per **Rechtsklick** auf die erste freie Zelle unterhalb des Spaltenkopfes ***"Kurve"*** kann eine an wählbare Kurve selektiert werden.



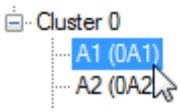


Alternativ dazu kann mit einem Doppelklick in der Kurvenspalte das Fenster **Base Line and Hysteresis** geöffnet werden. In diesem Dialogfenster können der Kanal, die Basislinie für allfällige Nulldurchgänge, die Hysterese dazu und das Cursor-Paar, das als Messfenster dient, festgelegt werden.

Sobald eine Kurve selektiert wurde, werden die Cursor-Werte ermittelt und in der Tabelle dargestellt.

Die Kurven, die in der **Skalarfunktions Tabelle** per Zeile ausgewählt werden können, sind im zugehörigen **Kurvenfenster** vorhanden.

Kurven können auf zwei verschiedene Arten mittels Drag & Drop in die Spalte *"Kurve"* gezogen werden:



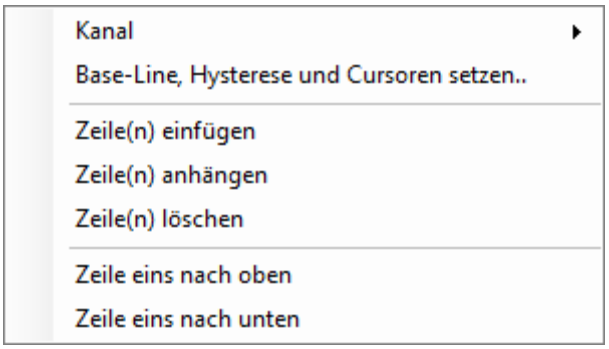
Vom Signalquellen Browser

vom Kanal-Schalter im Kurvenfenster

Die Skalarfunktion Tabelle wird normalerweise ausschliesslich im **Single Trigger-Mode** oder in einer Schleife einer Autosequenz mit entsprechender Wartezeit aktualisiert.

Falls Auto-Refresh aktiviert ist, so wird die Tabelle auch dann aktualisiert, wenn Single Trigger Mode ausgeschaltet ist.

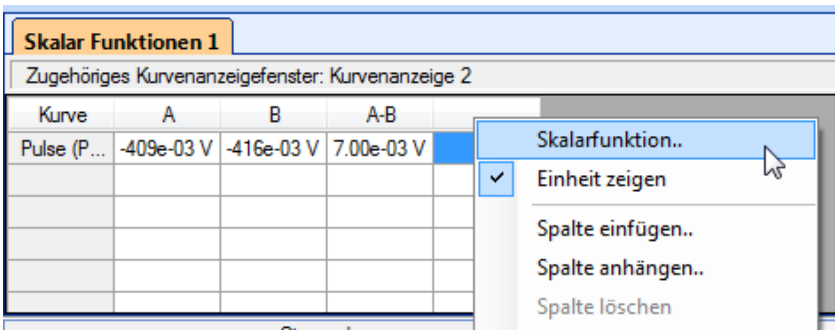
Die Basislinie und die Hysteresis müssen bei Bezugspegel-basierenden Skalaren gesetzt werden.



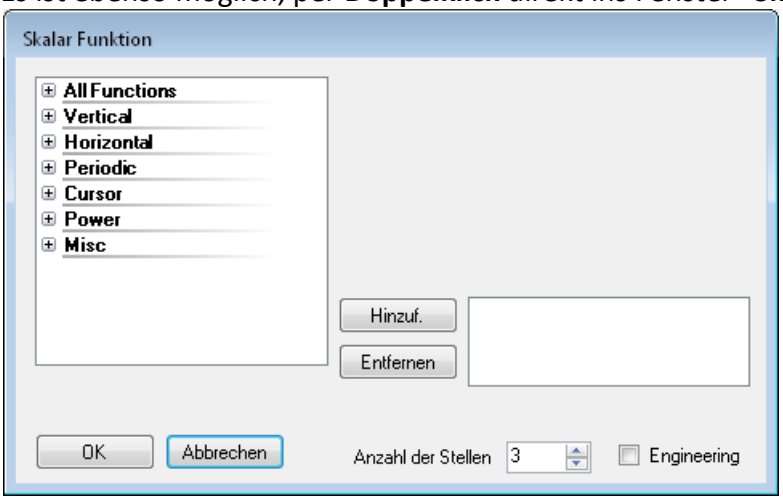
Im **Kurven-Kontextmenü** (rechte Maustaste) können **neue Zeilen** eingefügt, gelöscht oder angehängt werden. Auch die Reihenfolge der Zeilen lässt sich ändern mit *Zeile eins nach oben* und *Zeile eins nach unten*.

19.1.2 Auswahl der Skalarfunktion

Eine Skalarfunktion kann auf verschiedene Arten einer Kolonne zugewiesen werden. Eine Möglichkeit ist via Rechtsklick auf die Kopfzeile einer Kolonne das Kontextmenü öffnen und *Skalarfunktion..* auswählen.



Es ist ebenso möglich, per **Doppelklick** direkt ins Fenster **"Skalar Funktion"** zu gelangen.



Die **Skalarfunktionen** sind in verschiedene **Kategorien** unterteilt:

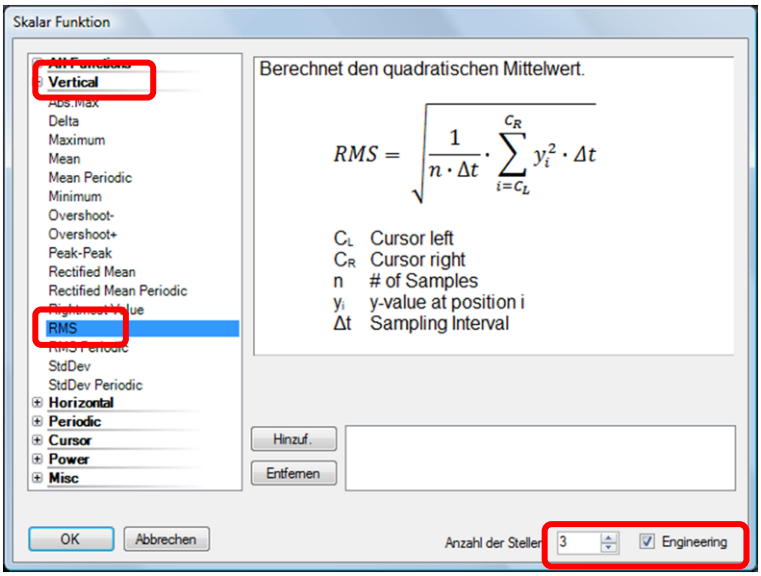
- **All Functions:** Listet alle Skalarfunktionen auf.
- **Vertical:** Listet die Skalarfunktionen auf, die vertikale Parameter der Kurve messen.
- **Horizontal:** Listet die Skalarfunktionen, die horizontale Parameter der Kurve berechnen. Diese Skalars sind typischerweise Zeitmessungen und benötigen meistens als Zusatzinformation die Basislinie und Hysterese.
- **Periodic:** Listet die periodischen Skalarfunktionen auf. Diese Skalare können nur auf periodische Signale angewendet werden. Sie benötigen als Zusatzinformation die Basislinie und Hysterese für die Ermittlung der ganzen Perioden.
- **Cursor:** Listet die Skalarfunktionen auf, die sich auf Cursor-Messungen beziehen.
- **Power:** Listet die Skalarfunktionen auf, die sich auf elektrische Leistungsmessungen beziehen.
- **Misc:** Listet spezielle Skalarfunktionen auf wie "Area" oder "Number of Triggers", die keiner der oben genannten Kategorien angehören.

Einige Skalarfunktionen werden bezogen auf eine bestimmte Basislinie mit Hysterese ausgewertet. Im Kapitel Beschreibung der Skalarfunktionen sind die Informationen betreffend Abhängigkeit der Basislinie angegeben.

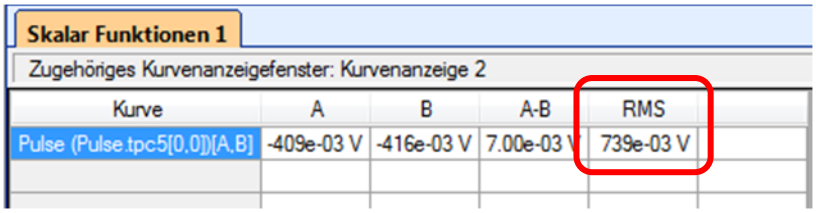
19.1.3 Beispiel 1: RMS

- *RMS* - Vertikale Skalarfunktion; ohne Referenzkanal; ohne Basislinie und Hysterese

Doppelklicken Sie auf die **Kopfzeile** einer Kolonne, um die aktuelle Skalarfunktion zu ersetzen. Klicken Sie im Dialog "Skalar Funktion" auf die **Funktions-Gruppe "Vertical".**
Selektieren Sie die **Funktion "RMS"**. Die Berechnungs-Formel mit Beschreibung und Legende wird im Fenster rechts der Liste angezeigt. Zur Darstellung des Messwertes wird **Engineering Mode** angewählt mit **drei Nachkommastellen**. TranAX merkt sich diese Einstellung und wird diese als Standard für die nächsten Funktionen verwenden.
Klicken Sie anschliessend "OK" um die Einstellungen zu akzeptieren und die Funktion "RMS" in die Tabelle einzufügen. Die vorherige Skalarfunktion wird ersetzt.



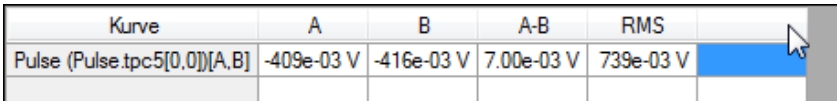
Der **RMS-Wert** wird aus der angewählten Kurve zwischen den **Cursor A und B** berechnet und in der **Spalte "RMS" angezeigt**. Rechts der Spalte "RMS" wurde automatisch eine **neue Spalte angehängt**, um das Hinzufügen einer weiteren Funktion zu erleichtern.



Normalerweise werden die **Skalar-Berechnungen** zwischen den selektierten **Cursors durchgeführt** (siehe Beschreibung der Skalarfunktionen). Unter Kanalauswahl wurde beschrieben, wie die Cursors ausgewählt werden. Im zugehörigen Kurvenfenster lassen sich die Cursor visualisieren und platzieren, womit das Messfenster pro Zeile definiert wird. Durch **unterschiedliche Cursor-Paare** kann das Messfenster pro **Zeile individuell definiert** werden.

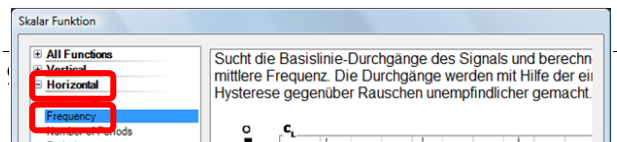
19.1.4 Beispiel 2: Frequency

- *Frequency* - Horizontale Skalarfunktion; ohne Referenzkanal; mit Basisline und Hysterese



Doppelklick auf die freie Kolonnen-Kopfzeile um das Fenster **Skalar Funktion** zu öffnen.

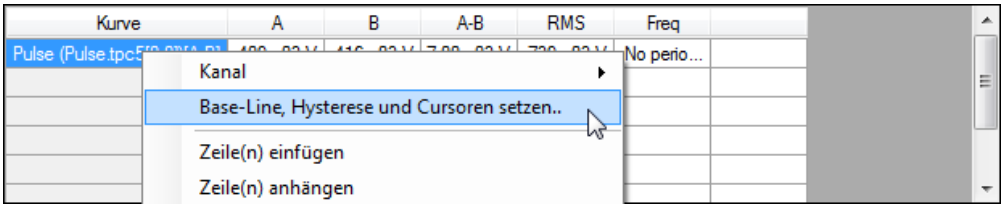
Anschliessend öffnen Sie die Liste **"Horizontal"** und selektieren die **Funktion "Frequency"**. Die Berechnungsformel und eine Grafik werden rechts der Liste angezeigt. Der Anzeige-Mode ist nach wie vor Engineering mit drei Nachkommastellen.
Klicken sie den Knopf "Hinzuf.", um den Skalar "Frequency" in die Liste des ausgewählten Skalars beizufügen.



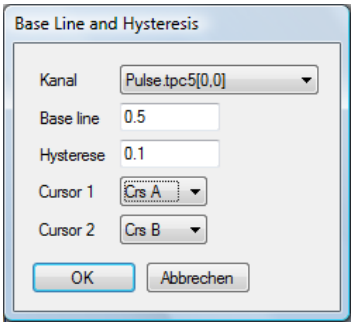
Es können auch weitere Funktionen ausgewählt und der Liste hinzu geführt werden. Jeder Eintrag wird in einer eigenen Spalte bei den Skalaren dargestellt.

Klicken Sie "OK", damit die eingetragenen Skalare, im Beispiel nur "Frequency", in der Skalarfunktions-Tabelle ganz rechts hinzugefügt werden. Bei mehreren Einträgen in der Liste werden die Skalare der Reihe nach in der Tabelle von links nach rechts angehängt.

Tätigen Sie einen Rechtsklick auf die aktive Zeile in der Kolonne *Kurve* und klicken Sie "Base-Line, Hysterese und Cursor setzen..", damit sich der Dialog "**Base Line und Hysterese**" öffnet. Alternativ dazu doppelklicken auf den gewünschten Kanal.

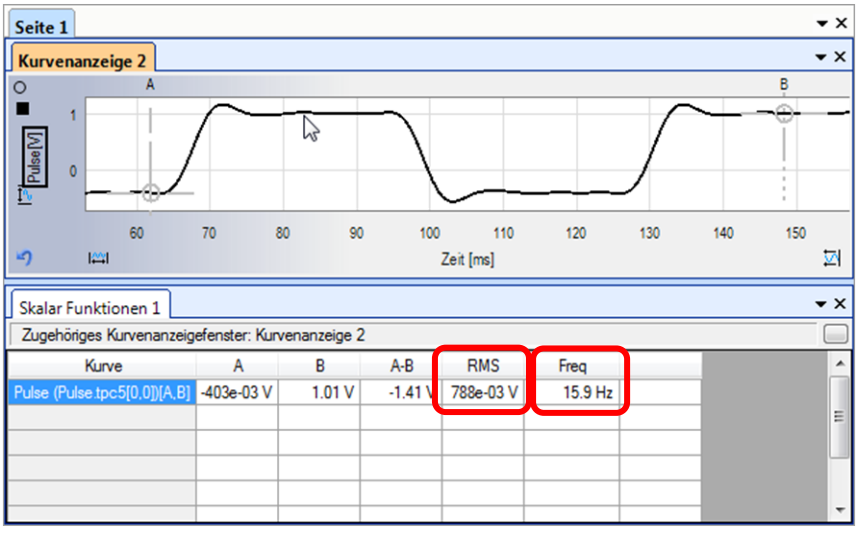


Im "**Base Line and Hysteresis**" Dialog kann nun die Messkurve, die Basislinie, die Hysterese und das Messfenster, definiert durch die Cursor, in dem zugehörigen Kurvenfenster bestimmt werden.



Die Frequenz wird nun bei der Basislinie von 0.5 V mit einer Hysterese von 0.1 V zwischen den Cursor A und B ermittelt.

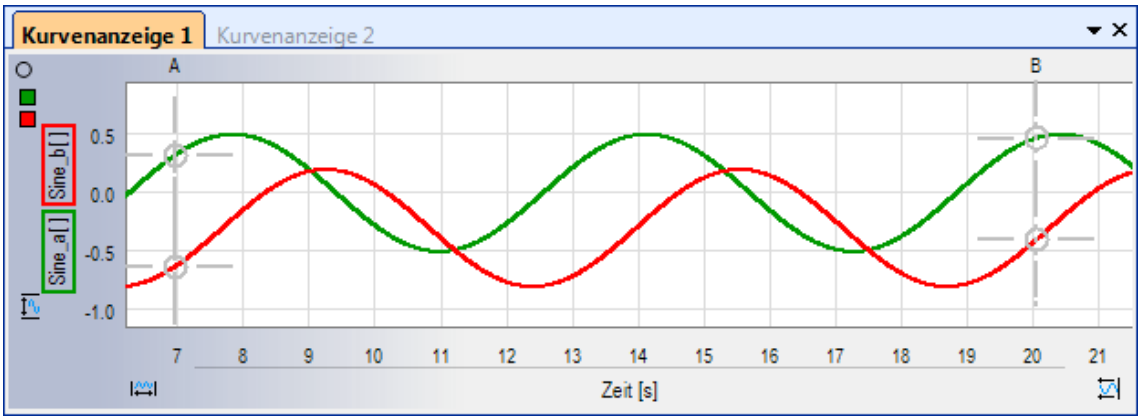
In der Tabelle Skalar Funktionen 1 können nun die Ergebnisse entnommen werden.

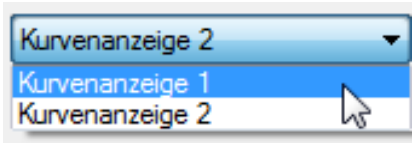


19.1.5 Beispiel 3: Phase

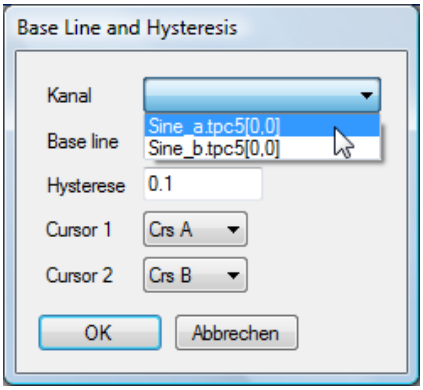
- *Phase* - Horizontale Skalarfunktion; mit Referenzkanal; mit Basislinie und Hysterese

In der folgenden Kurvenanzeige 1 sind zwei Kurven vorhanden, deren Phasenlage zu messen ist.



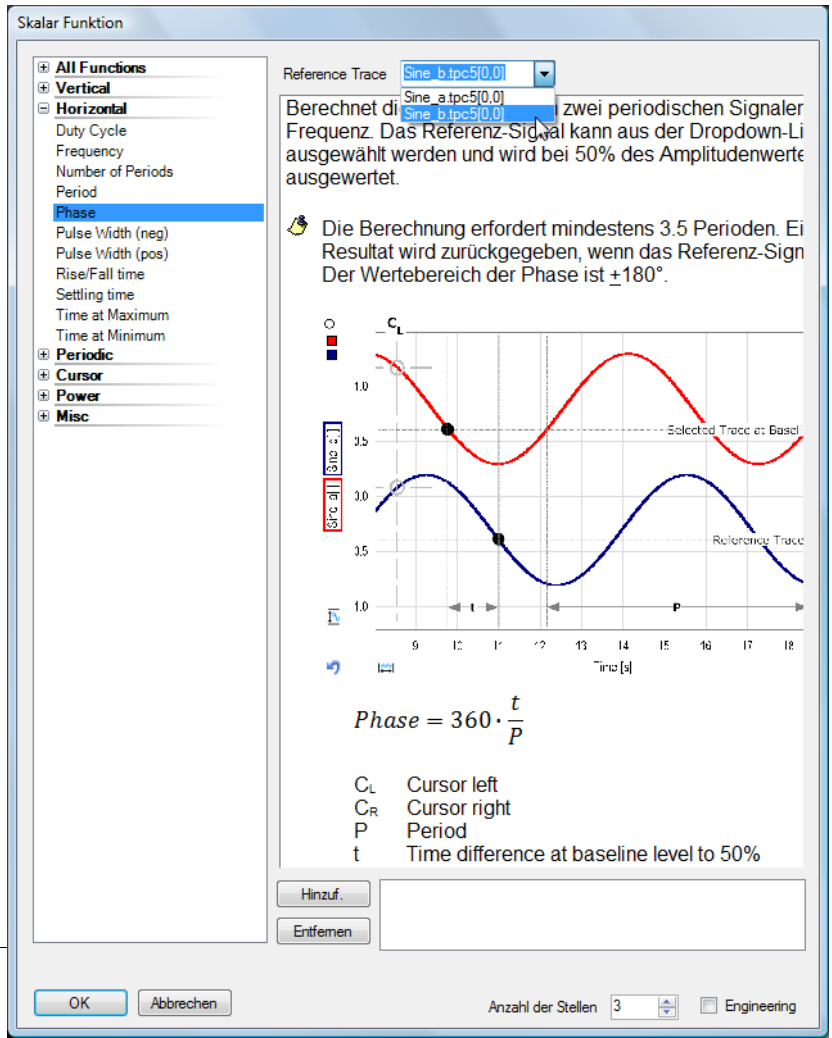


Zuerst muss die **Skalartabelle** auf die zugehörige **Kurvenanzeige eingestellt** sein. Wie weiter oben erklärt muss dazu bei der Auswahl "Zugehöriges Fenster" die entsprechende Kurvenanzeige ausgewählt werden. In diesem Fall Kurvenanzeige 1.



Sobald die Zugehörigkeit auf die "Kurvenanzeige 1" eingestellt ist, kann in der Skalartabelle auf deren Kurven zugegriffen werden.
Doppelklick auf die **erste Zeile** in der Kanal-Spalte. Das Fenster **Base Line and Hysteresis** wird geöffnet. Der Kanal *Sine_a* wird ausgewählt als Berechnungsquelle der ersten Zeile der Skalartabelle.

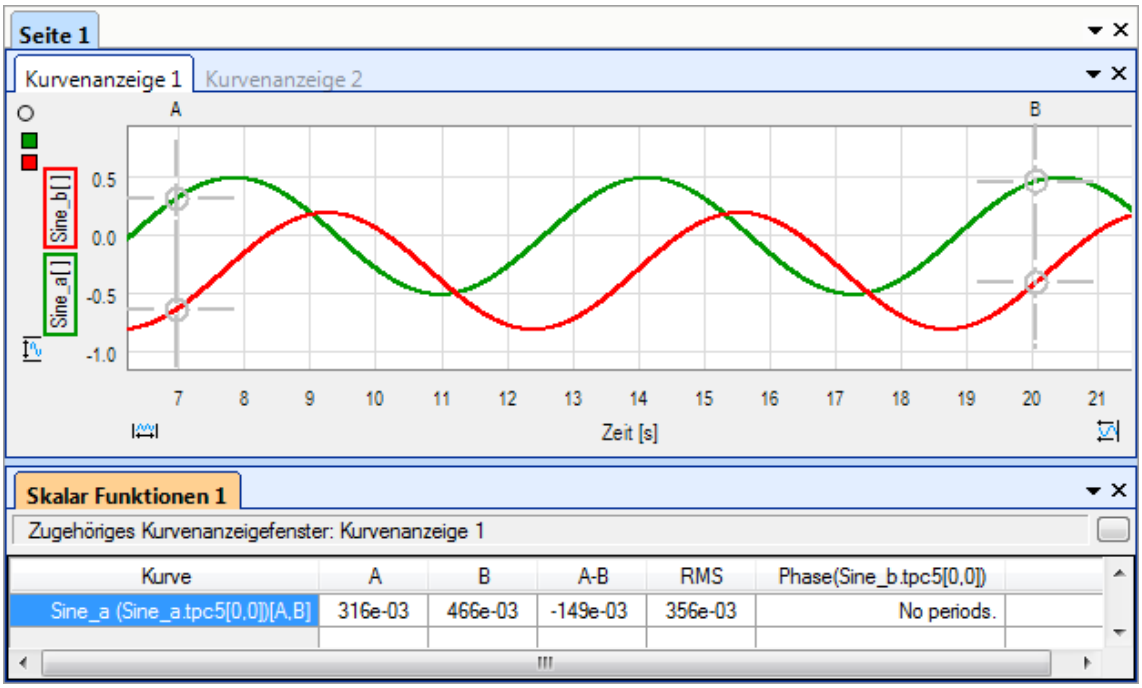
In diesem Beispiel soll der eingestellte Skalar "Frequency" durch "Phase" ersetzt werden. Dazu **Doppelklicken** Sie auf die Spalte mit der Skalarfunktion "Frequency". Wählen Sie in der **Liste "Horizontal" die Funktion "Phase"**. Selektieren Sie den **Referenzkanal "Sine_b"**, zu dem die Phase berechnet werden soll.

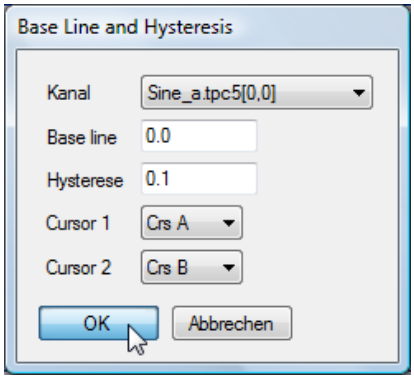


Das ist der Bezugskanal zum bereits eingestellten Kanal "Sine_a".

Mit Klick auf **OK** wird die Einstellung übernommen und das Fenster **Skalar Funktion** geschlossen.

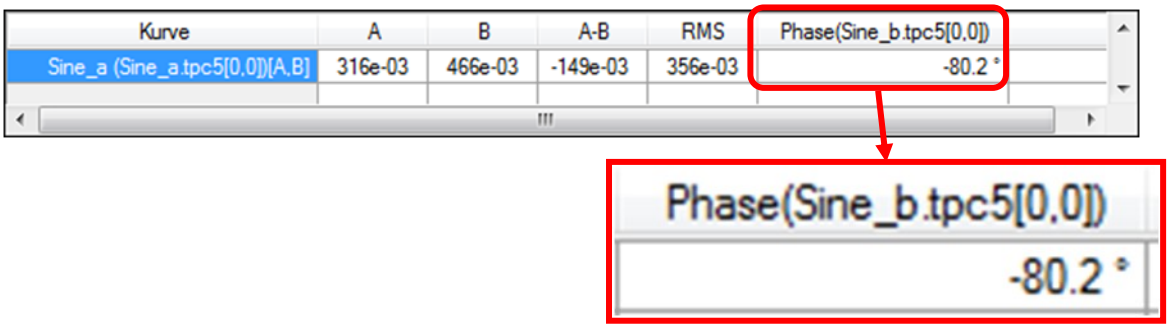
Die Phase gibt das Messergebnis "No periods" zurück. Das hat damit zu tun, dass entweder **nicht genügend Perioden** im Messfenster liegen oder dass der Basislinien-Pegel **keine Durchgangspunkte** mit der Kurve findet. Beim näheren Hinschauen stellt man fest, dass die Basislinie zur Berechnungen an der Kurve *Sine_a* idealerweise bei 0 V liegt.



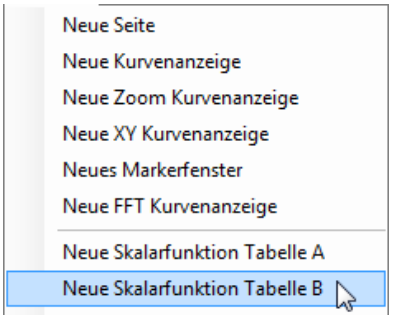


Im Fenster **"Base Line and Hysteresis"** werden die Werte, wie im nächsten Bild zu sehen ist, eingestellt.
Die Hysterese sollte nie auf null eingestellt werden, weil dadurch eventuell durch Rauschen oder Störungen zusätzliche Nulldurchgänge vorgetäuscht werden können. Klicken Sie **"OK"** zum Akzeptieren der Werte und Schliessen des Fensters.

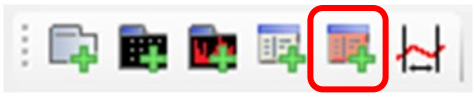
Nun sind Basislinien-Durchgänge vorhanden und die Phase kann berechnet werden.



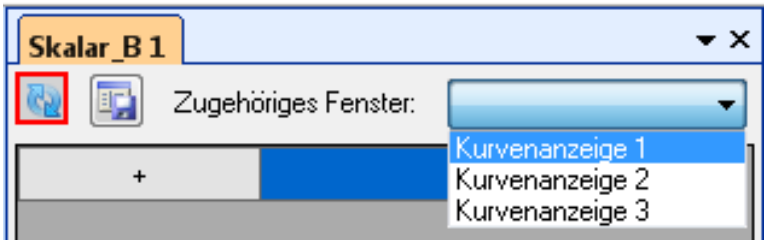
19.2 Skalarfunktionen Tabelle B



Eine neue Skalarfunktions-Tabelle B kann im Menü ***"Ansicht" / "Neue Skalarfunktion Tabelle B"*** geöffnet werden oder mit einem Klick auf das Icon "**Neue Skalartabelle B**".

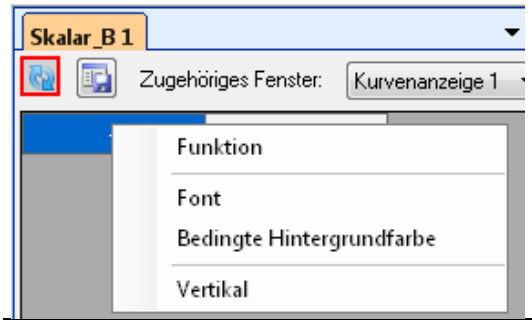


Das zugehörige Kurvenfenster, also die Quelle der zur Verfügung stehenden Kurven, kann im Dialog "**Zugehöriges Fenster**" ausgewählt werden. Hier werden alle verfügbaren Kurvenanzeigen aufgelistet.



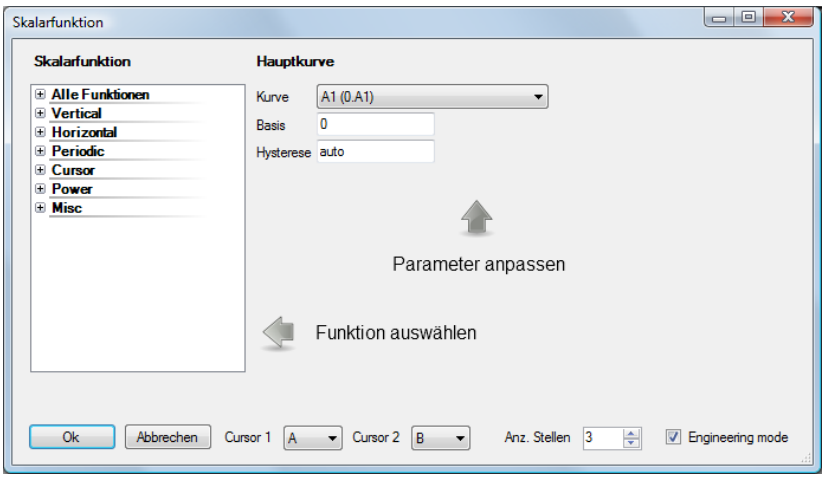
Falls gewünscht kann die neu erstellte Tabelle nun sehr **flexibel positioniert** werden. Durch Ziehen des Registerblattes ***"Skalar_B 1"*** lässt sie sich auf einer beliebigen Seite des aktiven Kurvenfensters platzieren.

19.2.1 Skalarfunktion hinzufügen



Um der Tabelle eine **Funktion hinzuzufügen** kann mit **Rechts-Klick** auf das **+** -Symbol das Kontextmenü geöffnet werden. Mit der Auswahl Funktion wird anschliessend das Fenster Skalarfunktion geöffnet. Alternativ dazu öffnet sich das Fenster Skalarfunktion direkt mit einem **Doppelklick** auf das **+** -Symbol in der untersten Linie bzw. in der Kolonne ganz rechts.

Die Kurven, die in der Skalarfunktions Tabelle per Zeile bzw. per Kolonne ausgewählt werden können sind im zugehörigen Kurvenfenster vorhanden.

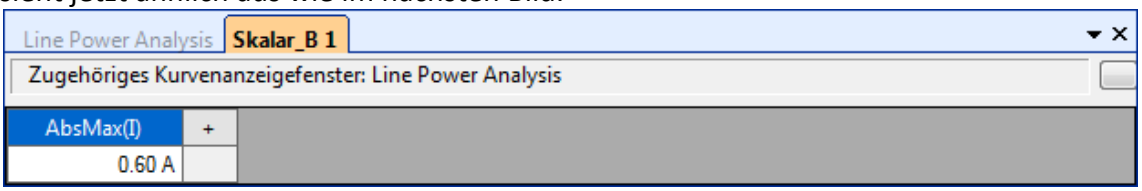


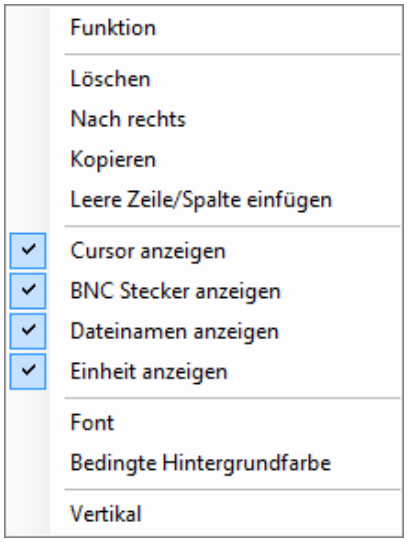
In diesem Fenster werden alle **Einstellungen** vorgenommen, die zu einer bestimmten **Skalarfunktion** gehören. Die Funktionen sind unterteilt in folgende Kategorien (wie bei Skalar-Tabelle A):

- **All Functions:** Alle Funktionen sind aufgelistet.
- **Vertical:** Aufgelistet sind diejenigen Funktionen, die vertikale Messungen an der Kurve durchführen.
- **Horizontal:** Aufgelistet sind diejenigen Funktionen, die horizontale Messungen an der Kurve durchführen. Diese Skalarfunktionen ermitteln typischerweise zeitabhängige Messgrössen wie Pulsbreite oder die Periodendauer. In der Regel benötigen diese Skalarfunktionen eine bestimmte Schwelle (Baseline) mit Hysterese.
- **Periodic:** Listet die periodischen Skalarfunktionen auf. Diese Skalare können nur auf periodische Signale angewendet werden. Sie benötigen als Zusatzinformation die Basislinie und Hysterese für die Ermittlung der ganzen Perioden.
- **Cursor:** Listet die Skalarfunktionen auf, die sich auf Cursor-Messungen beziehen.
- **Power:** Aufgelistet sind diejenigen Funktionen, die üblicherweise zu elektrischen Leistungsmessungen eingesetzt werden.
- **Misc:** Hier werden einige Spezial-Skalarfunktionen wie *Area* oder *Number of Triggers* aufgelistet, die keiner der oben erwähnten Kategorien zugeordnet werden können.

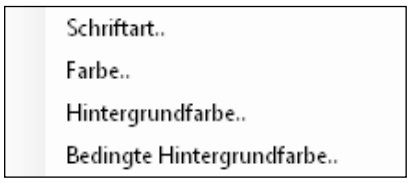
Nun wird die **Hauptkurve** für die **Berechnung** ausgewählt. Einige Funktionen wie die Scheinleistung (Apparent Power) beziehen sich noch auf eine zweite Kurve, die zugehörige Referenzkur-

ve. Die Einstellungen für die Basislinien mit Hysterese legen die Pegel zur Ermittlung von ganzen, abgeschlossenen Perioden fest.
Es muss weiter **definiert** werden, zwischen **welchen Cursor** die Skalarfunktion berechnet werden soll und in welchem Zahlenformat.
Mit Klick auf **OK** wird das Fenster geschlossen und die Einstellungen übernommen. Die Tabelle sieht jetzt ähnlich aus wie im nächsten Bild.





Ein **Rechts-Klick** auf die Zelle mit der **Skalarfunktion** in der Tabelle öffnet ein Dialogfenster. Das ermöglicht die Auswahl einer **anderen Skalarfunktion**, eine Kopie der aktuellen Funktion zu erstellen, die aktuelle Funktion zu löschen oder eine neue Spalte/Zeile einzufügen.
Weiter ist es möglich, die Darstellungsweise des Resultates anzupassen. Wahlweise können die Cursors, die Einheit, die Signaleingangsbuchse, und der Dateiname ein- oder ausgeblendet werden.
Die Auswahl ***Vertikal*** konvertiert die Tabelle von Zeilendarstellung nach **Spaltendarstellung**.
Die vertikale Darstellung eignet sich, wenn die Tabelle auf einer Seite des Kurvenfensters angedockt ist.



Ein Rechts-Klick auf eine Tabellenzelle, in der das Messergebnis dargestellt wird, öffnet ein Dialogfenster zur Konfiguration der Visualisierung des Resultates. Es können auch mehrere Zellen markiert werden und dann gemeinsam angepasst werden.

Ein Rechts-Klick auf die Zelle mit dem "+" öffnet einen Dialog zur Auswahl einer Skalarfunktion und bietet die Möglichkeit die Ausrichtung zu ändern (horizontal/vertikal).

Die Skalarfunktions Tabelle wird **normalerweise** ausschliesslich im **Single Trigger** Mode oder in einer Schleife einer Autosequenz mit entsprechender Wartezeit aktualisiert. Es besteht aber die Möglichkeit, Auto-Refresh der Tabelle einzuschalten.

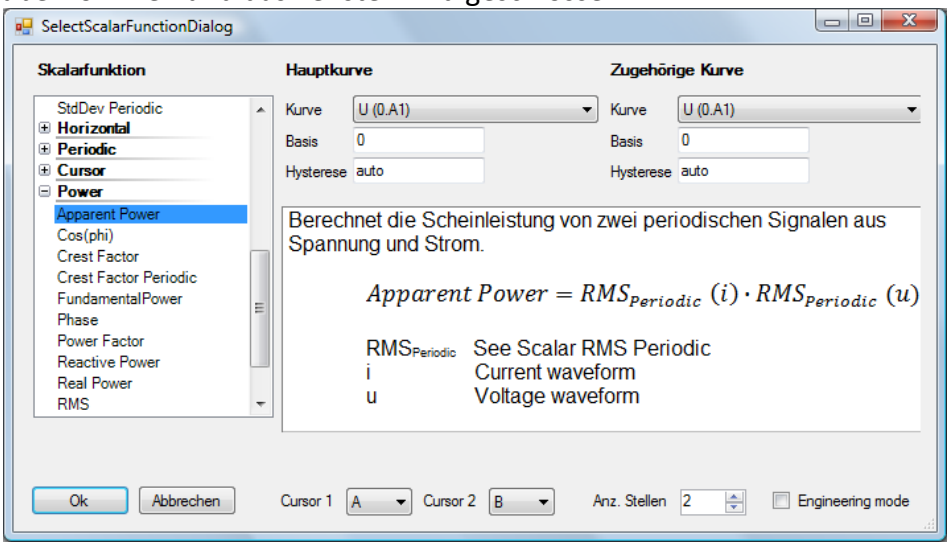
19.2.2 Beispiel 1: Apparent Power

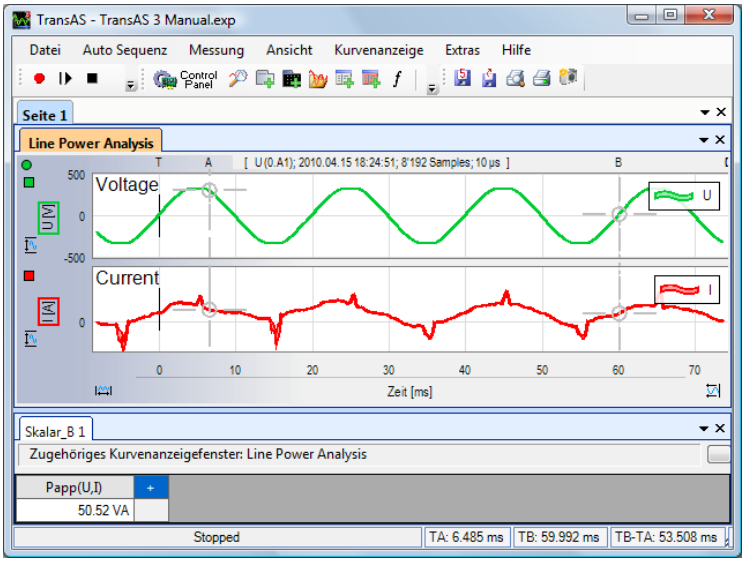
- *Apparent Power* (Scheinleistung) - Kategorie Power; mit zugehöriger Referenzkurve; mit Basislinie und Hysterese

Tätigen Sie einen Doppel-Klick auf die Skalartabelle um entweder eine neue Funktion zu erstellen oder eine bestehende zu ersetzten. Öffnen Sie die Kategorie Power und wählen sie die Funktion "Apparent Power". Die Berechnungsformel wird im Fenster rechts der Liste angezeigt.

Wählen Sie die **Hauptkurve** aus (davon ausgehend, es sei eine Spannungskurve) mit Basislinie bei 0V und einer automatisch festgelegten Hysterese, welche 10% des Spitzen-Spitzenwert der Kurve entspricht.
Nun wählen sie die **Referenzkurve** aus (davon ausgehend, es sei eine Stromkurve) mit Basislinie bei 0V und einer automatisch festgelegten Hysterese, welche 10% (+/- 5%) des Spitzen-Spitzen-Wertes der Kurve entspricht.

Nach Grundeinstellung werden die Resultate im Nicht-Engineering-Mode dargestellt. In diesem Fall zwei Nachkommastellen beibehalten. Gegebenenfalls könnte hier die Anzahl Stellen angepasst und/oder auf Engineering-Format gewechselt werden. Mit Klick auf **OK** werden die Werte übernommen und das Fenster wird geschlossen.





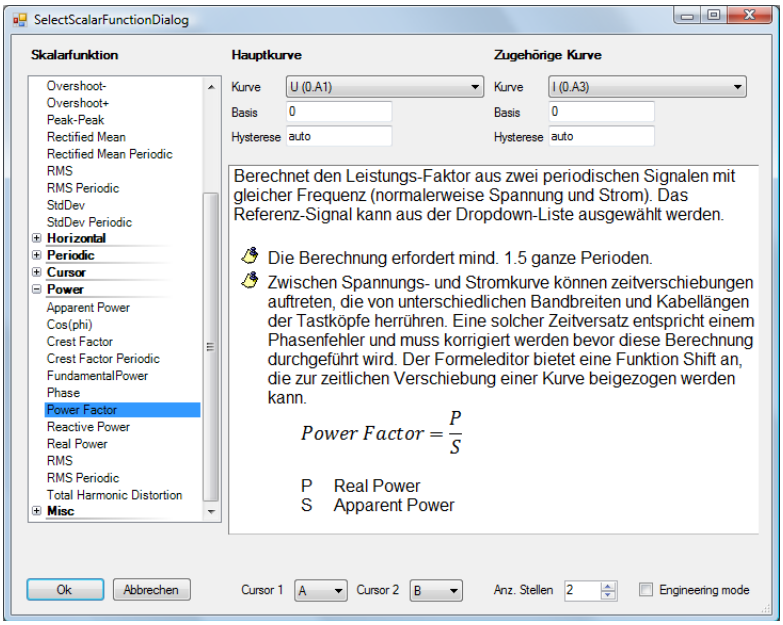
Die Scheinleistung wird nun zwischen den Cursor A und B berechnet und das Resultat in der Spalte "Papp(U/I)" dargestellt. Rechts von dieser Spalte wurde automatisch eine neue Spalte eingefügt.

Die Skalarfunktionen werden zwischen den gewählten Cursor berechnet (siehe auch Beschreibung der Skalarfunktionen). In der zugehörigen Kurvenanzeige können die gewählten Cursors so platziert werden, dass der gewünschte Kurvenbereich als Messfenster für die Berechnung selektiert ist.

19.2.3 Beispiel 2: Power Factor

- *Power Factor* - Kategorie Power; mit zugehörigem Referenzkanal und Basislinie mit Hysterese

Nun wird eine weitere Skalarfunktion hinzugefügt. Doppelklicken Sie auf das **+** -Symbol. Wählen Sie im Dialog "Skalarfunktion" die Kategorie "Power" öffnen und selektieren Sie die Funktion "Power Factor".
Das Zahlenformat ist immer noch Non-Engineering mit 2 Stellen, weil TranAX sich an die letzte Einstellung erinnert. Mit einem Klick auf OK werden die Einstellungen übernommen und das Fenster wird geschlossen.

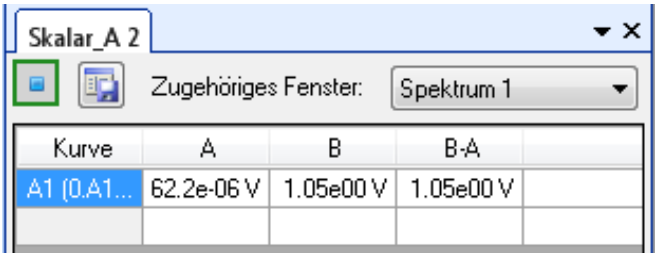


Der Leistungsfaktor aus den Kanälen *U(0.A1)* und *I(0.A3)* wird nun berechnet zwischen den Cursor A und B. Die Basislinien werden benötigt um Anfang und Ende von ganzen Perioden zu definieren. Die Hysteresis ist auf *"Auto"* gestellt, was einem Wert von 10% (+/- 5%) der Spitzen-Spitzen-Werte der jeweiligen Kurve entspricht.

Einige Skalarfunktionen benötigen eine bestimmte Schwelle (Baseline) mit Hysterese. Im Abschnitt Beschreibung der Skalarfunktionen werden alle Skalarfunktionen beschrieben und es ist ersichtlich, welche Einstellungsmöglichkeiten jede Funktion hat.

19.3 FFT Funktionen (Tabelle A/B)

Wenn beim Öffnen einer neuen Skalar-Tabelle (A oder B) ein **Spektrum-Fenster** den **Fokus** hat, wird automatisch die Skalar-Tabelle mit dem Spektrum als **"zugehörigem Fenster"** gebildet. Dabei werden auch automatische die vorhandenen Spektrum-Kurven in der Tabelle eingetragen.



Prinzipiell ist die Bedienung zu den Skalar-Tabellen von Spektren gleich, wie bei Skalar-Funktionen zu den zeitbezogenen Signalen. Einzig die Liste der möglichen **Funktionen** ist auf die Berechnung von **Spektrum-Parametern zugeschnitten**.

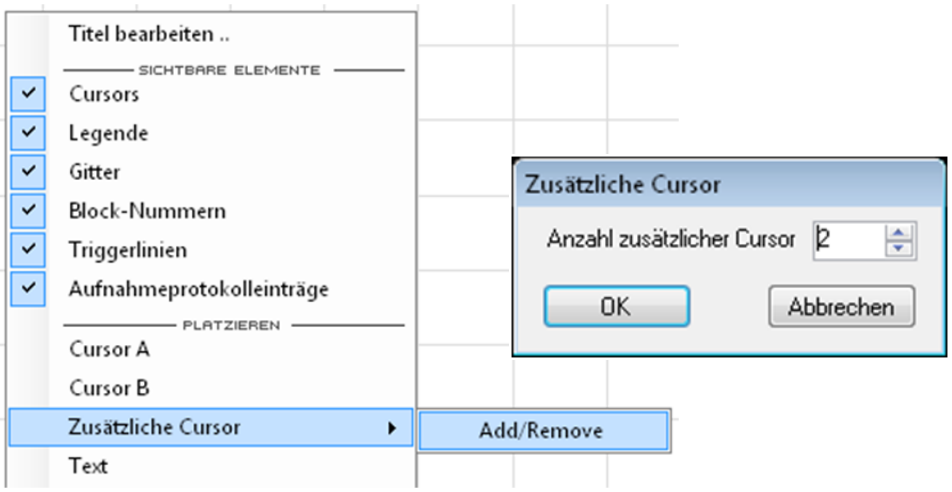
Folgende Spektrum-Parameter können berechnet werden:

- Cursor Amplitude
- Cursor Delta Amplitude
- Cursor Delta Position
- Cursor Position
- Cursor Ratio Amplitude (dB)
- Frequency at Maximum
- Maximum (zwischen zwei Cursors)

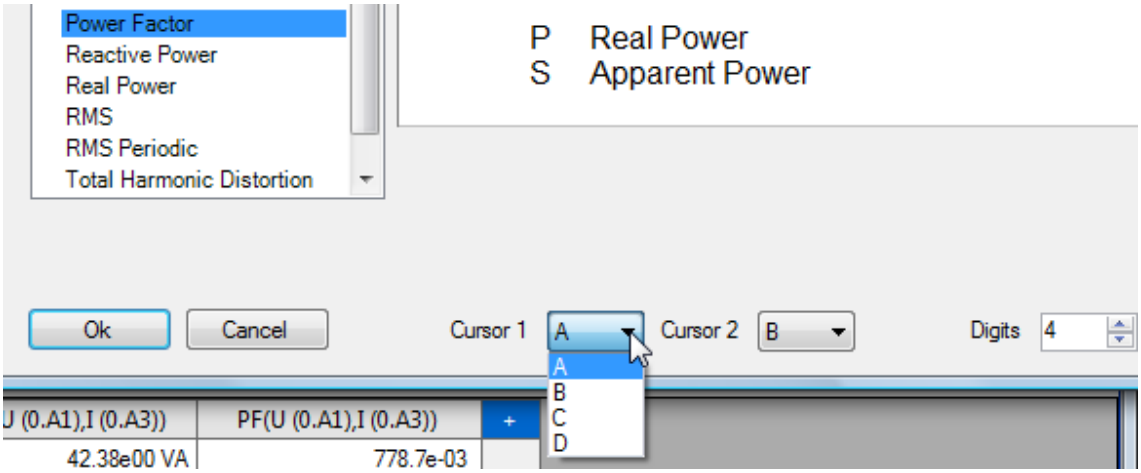
In den Skalar-Tabellen kann unter "Zugehöriges Fenster" von einem Zeitfenster (Kurvenanzeige) zu einem Spektrum (oder umgekehrt) gewechselt werden. Danach müssen jedoch die entsprechenden **Kurven neu via Drag & Drop** in die Tabelle gezogen werden. Auch müssen in der Regel die **Funktionen angepasst werden** ("Mean" z.B. kann in einem Spektrum nicht gerechnet werden).

19.4 Zusätzliche Cursors

Falls mehr als ein Messbereich, also **zusätzliche Cursor** benötigt werden, fügt man diese mit Rechtsklick im Kurvenfenster und der Auswahl *Zusätzliche Cursor* hinzu. Zusätzlich heisst zusätzlich zu den Cursor A und B.



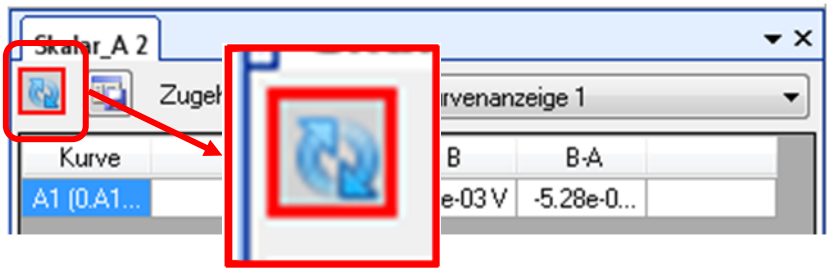
Falls die Anzahl zusätzlicher Cursors 2 ist, so werden anschliessend 4 Cursors im Kurvenfenster zur Verfügung stehen, C und D kommen zu A und B hinzu. Diese können dann im Fenster **Skalar Funktion** auch ausgewählt werden.



19.5 Auto-Refresh der Skalarfunktion Tabelle

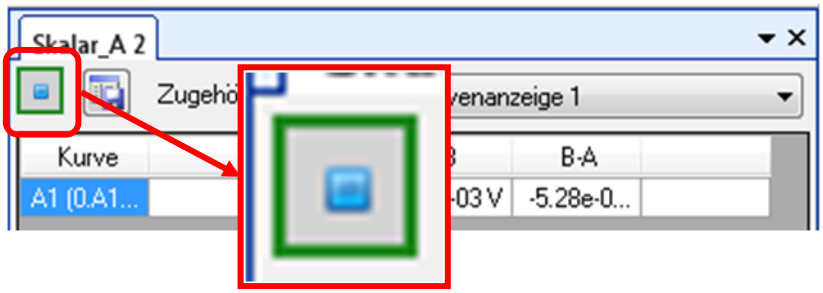
Im **Single Shot Mode** werden die Skalar-Werte der Skalarfunktions-Tabelle nach jeder Datenerfassung **aufdatiert**. Falls Single Shot Mode ausgeschaltet ist, werden die Skalar-Werte nur dann aufdatiert, wenn der **Schalter Auto-Refresh** eingeschaltet ist (grüner Rahmen).

Im folgenden Bild ist **Auto-Refresh ausgeschaltet (roter Rahmen)**.



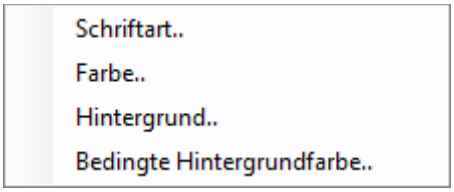
Da das Auto-Refreshing der Skalartabelle eine grosse **Auswirkung** auf die **Systemleistung** hat empfiehlt es sich, Auto-Refresh nur dann einzuschalten, wenn es wirklich benötigt wird. Falls die Update-Rate der Kurvenanzeige wichtiger ist als die Berechnungen der Skalarfunktionen sollte Auto-Refresh ausgeschaltet werden.

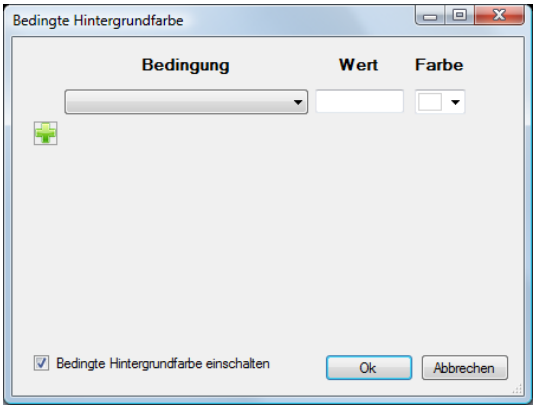
Im folgenden Bild ist **Auto-Refresh eingeschaltet (grüner Rahmen)**.



19.6 Bedingte Hintergrundfarbe

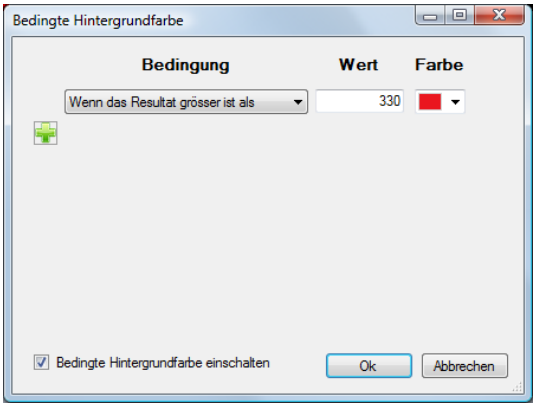
Zu jeder **Skalarfunktion** können **Bedingungen definiert** werden, die den Anwender bezüglich bestimmter Skalar-Werte **visuell** informieren. Mit einem **Rechts-Klick** auf den Wert der Skalarfunktion und der Auswahl "Bedingte Hintergrundfarbe" öffnet sich das entsprechende Fenster.



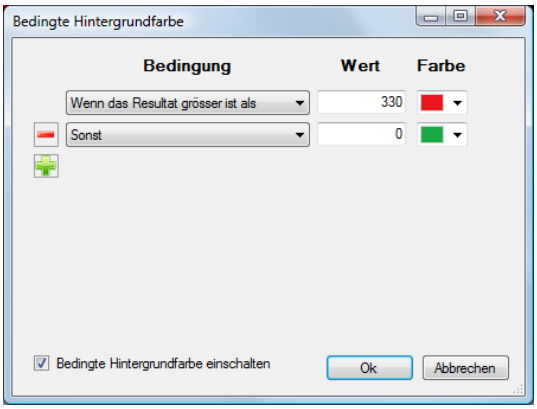


Für jedes **Feld** können **mehrere Bedingungen** definiert werden. Es stehen die folgenden Bedingungen zur Verfügung:

Wenn das Resultat grösser ist als
Wenn das Resultat kleiner ist als
Sonst
Funktion hat einen Fehler
Deaktiviert



Es wird nun die Bedingung definiert, dass die Hintergrundfarbe rot wird beim Überschreiten des Wertes von 330V.



Eine weitere Bedingung wird mit der Schaltfläche hinzugefügt.

Mit der zweiten Bedingung wird nur gesagt, dass die Hintergrundfarbe grün sein soll, falls die erste Bedingung nicht erfüllt ist.

Es wird empfohlen, die Werte in auf- oder absteigender Reihenfolge zu listen.

Die bedingte **Hintergrundfarbe** muss noch **aktiviert** und das Fenster mit **Ok** geschlossen werden. Die Skalarfunktion mit der bedingten Hintergrundfarbe wird nun entsprechend dem Skalar-Wert ausgewertet:

A(A1 (0.A1))	A(I (0.A3))	PP(A1 (0.A1)) [A,B]	+
23.1766 V	-0.0575 V	340e00 V	

Spannung ist **über 300V**, die Zelle wird **rot** gefärbt.

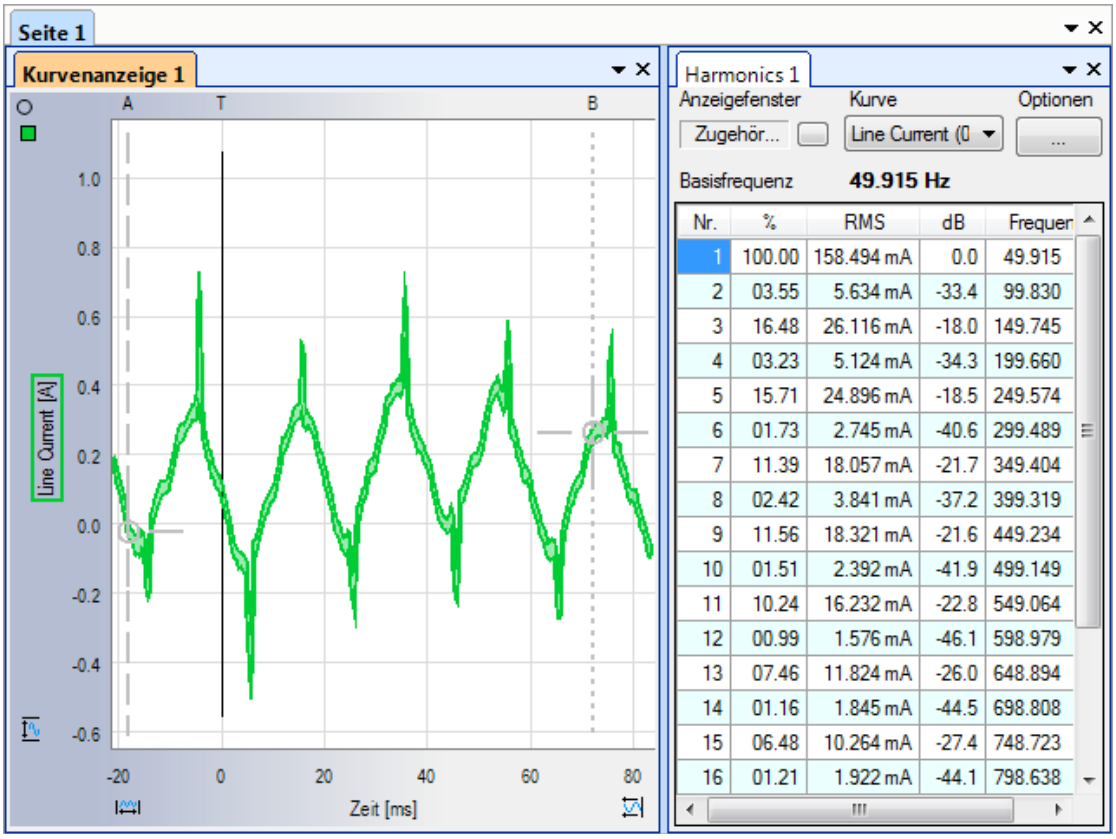
A(A1 (0.A1))	A(I (0.A3))	PP(A1 (0.A1)) [A,B]	+
23.1766 V	-0.0575 V	215e00 V	

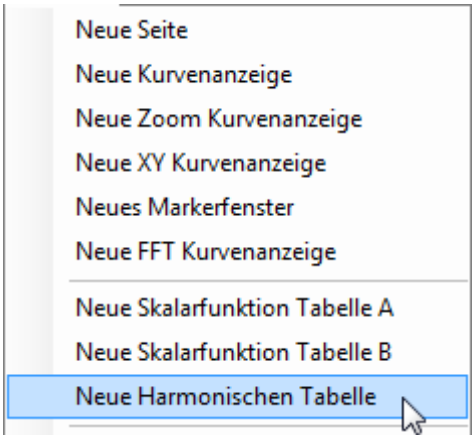
Bei einer Spannung **unter 300V** bleibt diese **grün**.

20 Tabelle der Harmonischen

Ein **periodisches Signal der Zeitebene** kann mit Hilfe der **Fourier Analyse** in die Frequenzebene konvertiert werden. Dabei werden die Amplituden bezogen auf die Frequenz aufgeschlüsselt in die Grundwelle und die Harmonischen.

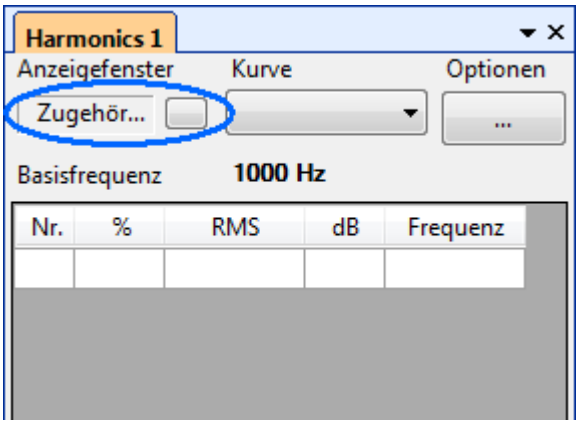
Die Tabelle der Harmonischen kann beliebig neben der Kurvenanzeige platziert oder als Registerkarte beibehalten werden. Für weitere Information betreffend der "Window Dock-in" Technik siehe auch Abschnitt TranAX Übersicht.

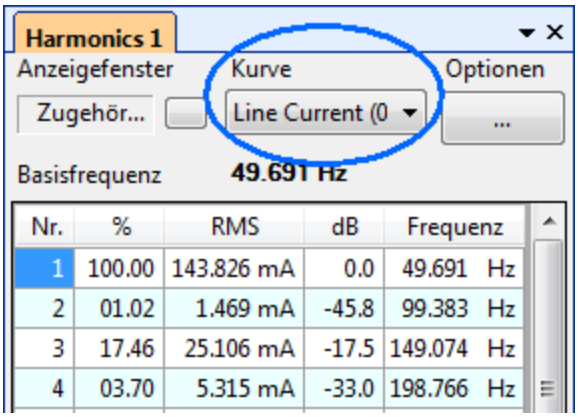




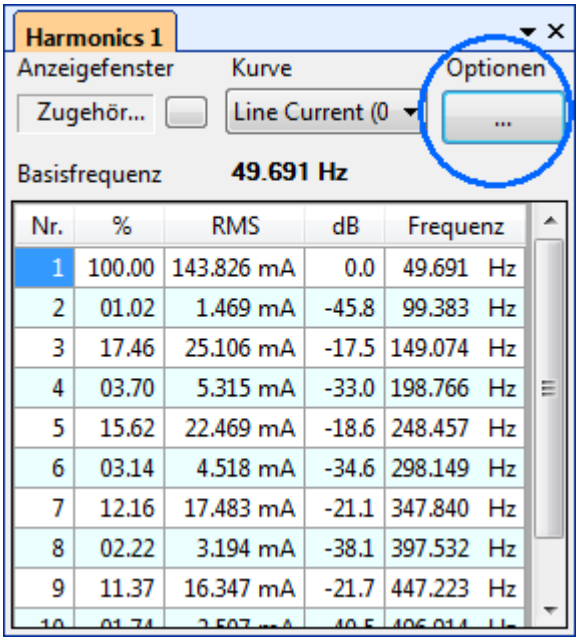
Eine neue Tabelle der Harmonischen wird im Menü ***"Ansicht" / "Neue Harmonischen Tabelle"*** geöffnet.

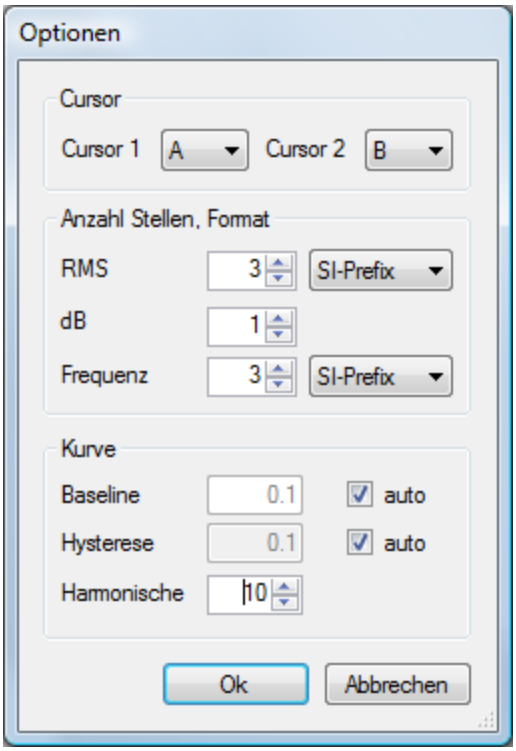
Eventuell muss dieser Tabelle zuerst eine **Kurvenanzeige zugeordnet** und anschliessend daraus die zu analysierende Kurve ausgewählt werden.





In den Optionen kann eingestellt werden, zwischen **welchen Cursors**, bei welcher Schwelle (Baseline) und Hysterese die Kurve **ausgewertet** werden soll.

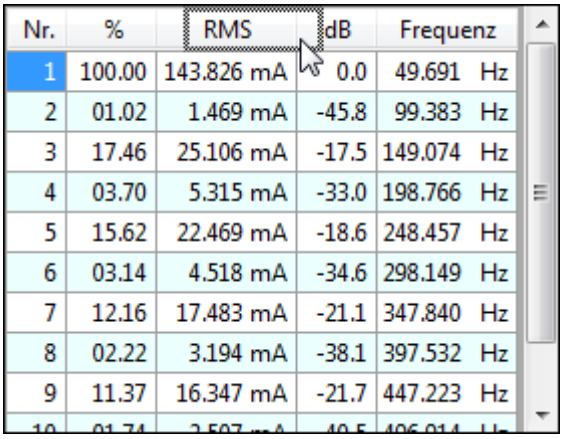




- **Cursor:** Die Harmonischen werden zwischen den Cursor A und B berechnet. Zur Berechnung müssen mindestens zwei Signalperioden zwischen den Cursor liegen.
- **Anzahl Stellen, Format:** Die Formate werden gewählt, wie die Resultate dargestellt werden sollen.
- **Kurve:** Die Schwelle (Baseline) und Hysterese sind für die Bestimmung der Periodendauer/Frequenz erforderlich. Falls **Auto** angewählt ist, wird die Basislinie in die Mitte des Signales und die Hysterese auf 10% (+/- 5%) des Spitzen-Spitzen-Wertes gelegt. Im Eingabefeld **Harmonische** wird eingetragen bis zu welcher Nummer die Harmonischen berechnet werden sollen (Bereich 1 ... 100).

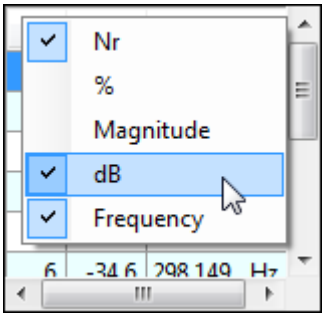
 © Elsys AG

Die Tabelle kann, hinsichtlich der Reihenfolge der Kolonnen und welche Kolonnen sichtbar sein sollen, angepasst werden.

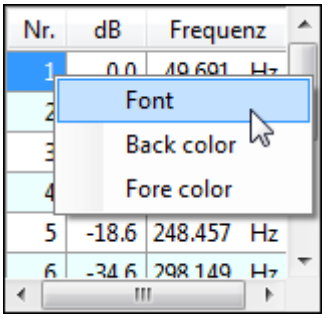


Nr.	%	RMS	dB	Frequenz
1	100.00	143.826 mA	0.0	49.691 Hz
2	01.02	1.469 mA	-45.8	99.383 Hz
3	17.46	25.106 mA	-17.5	149.074 Hz
4	03.70	5.315 mA	-33.0	198.766 Hz
5	15.62	22.469 mA	-18.6	248.457 Hz
6	03.14	4.518 mA	-34.6	298.149 Hz
7	12.16	17.483 mA	-21.1	347.840 Hz
8	02.22	3.194 mA	-38.1	397.532 Hz
9	11.37	16.347 mA	-21.7	447.223 Hz

Mit Hilfe von Drag & Drop kann die **Reihenfolge** der einzelnen Spalten **geändert** werden.



Via **Rechts-Klick** auf der Titellinie lassen sich Kolonnen ein- und ausschalten so dass nur die gewünschten **Informationen** angezeigt werden.

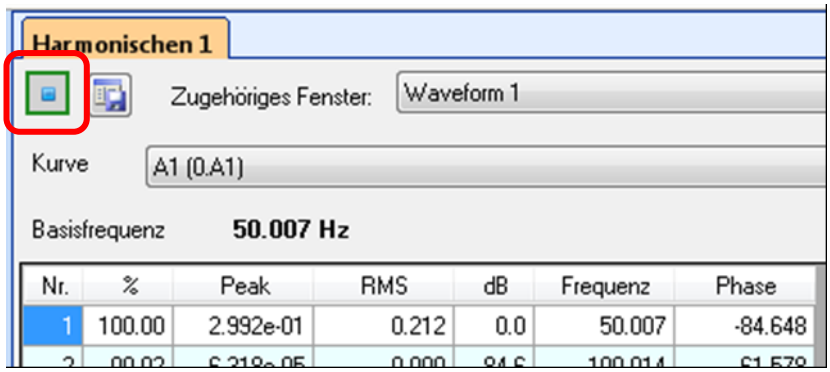


Rechts-Klick innerhalb einer **Zelle** oder einer Auswahl von Zellen um die Schrift und Farben zu editieren.

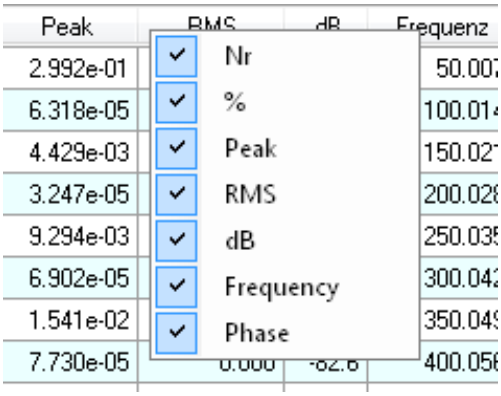
20.1 Erweiterte Harmonische Tabelle

Export Funktion: Durch Klicken des Knopfs kann die gesamte Tabelle im Text-Format exportiert werden. Diese Datei kann anschliessend z.B. in Excel wieder importiert und gegebenenfalls weiter bearbeitet und verwendet werden.

Aktualisierung während der Messung: Mit dem Knopf oben Links kann die Aktualisierung zur Laufzeit ein- oder ausgeschalten werden, gleich wie bei den Skalartabellen.



Beim hier dargestellten Zustand ist die Aktualisierung eingeschalten.

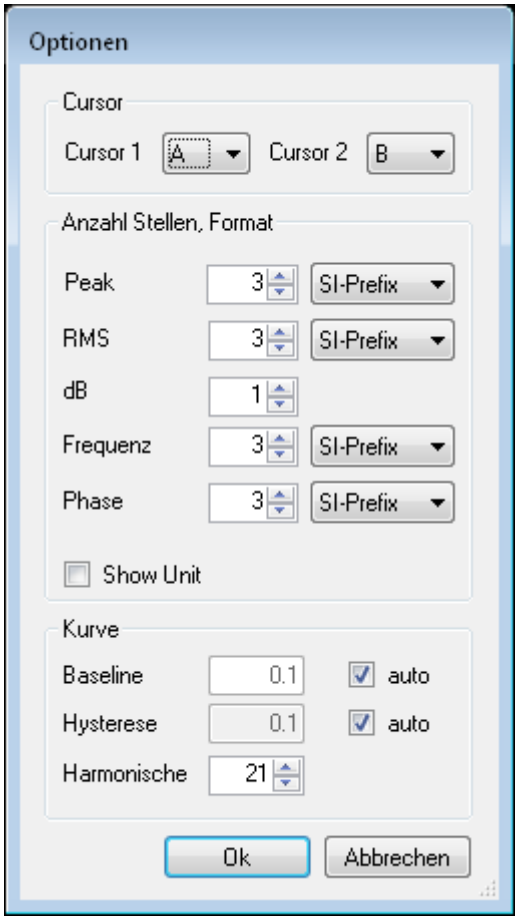


Mit einem **Rechtsklick** auf die obere Titelleiste können die einzelnen **Spalten ein- oder ausgeschalten** werden. Es werden nur die Spalten mit dem Häklein angezeigt.

Die Messwerte für **"Peak"** und die **"Phase"** der Harmonischen können nun ebenfalls in der Tabelle dargestellt werden.

Es ist darauf zu achten, dass die **Cursors richtig platziert** wurden. Um die Harmonischen berechnen zu können, werden **mindestens zwei ganze Perioden** der Grundwelle benötigt!

Bei den **"Optionen"** können die folgenden Einstellungen getätigt werden:

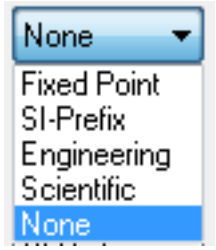


Auswahl der zu verwendenden Cursors: Als Standard sind A und B eingestellt.

Anzahl Stellen und Format für jede einzelne Spalte kann die Anzahl der Stellen und die Formatierung der Zahl gewählt werden.

Show Unit: Mit dieser Option kann bestimmt werden, ob die Einheiten bei den Resultat-Werten angezeigt werden oder nicht. Für Exporte zur weiteren Bearbeitung in externen Programmen wird diese Option mit Vorteil deaktiviert, so dass nur reine Zahlenwerte gespeichert werden.

Kurve: Baseline und Hysterese kann entweder manuell vorgegeben oder (wie empfohlen) automatisch ermittelt werden. Stark gestörte Signale erfordern oft eine manuelle Einstellung, vor allem der Hysterese.
Die Anzahl der zu berechnenden Harmonischen kann ebenfalls eingegeben werden.



Zu jeder Resultat-Spalte kann die Darstellung des Zahlenwerts definiert werden. Unterschieden wird zwischen **Fixed Point**, **SI-Prefix**, **Engineering**, **Scientific** und **None**.

Beispiel: Messwert für "Peak" beträgt 23'070.20897 V

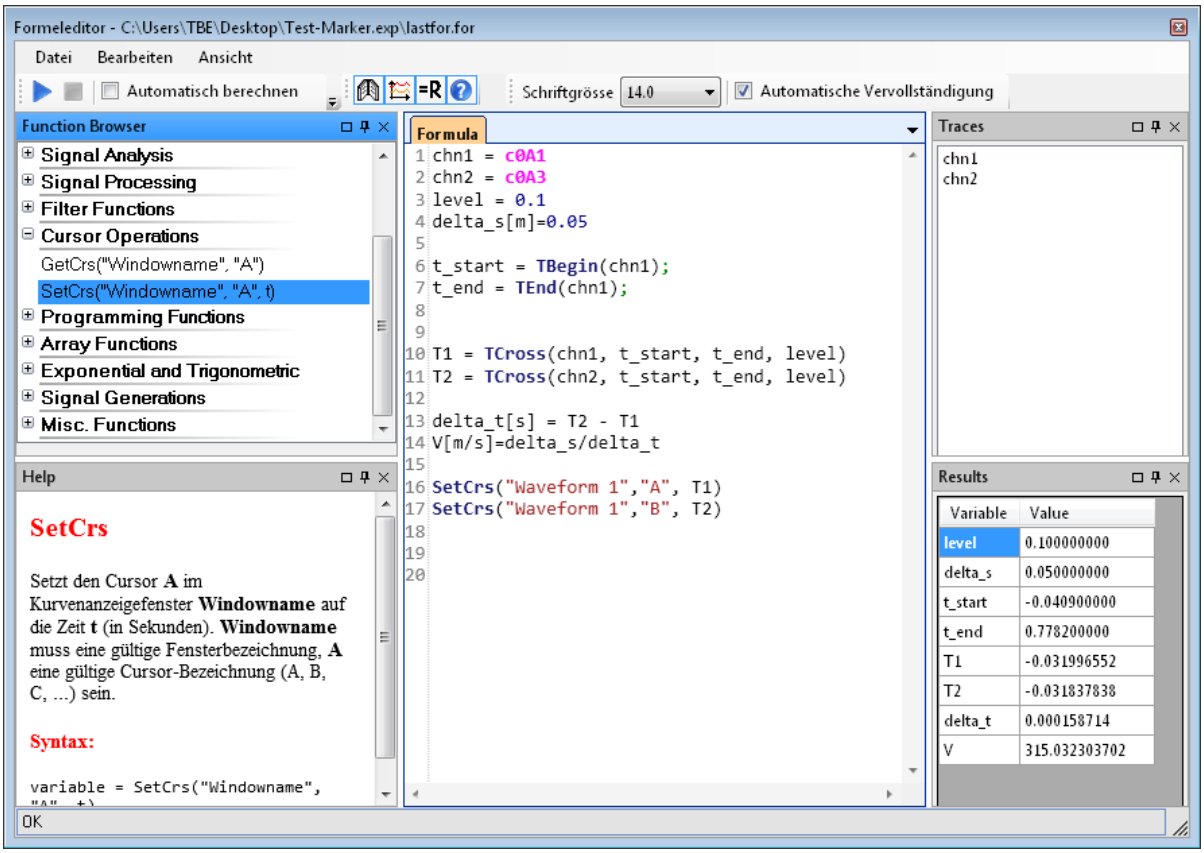
Format	Stellen	Anzeige
Fixed Point	3	23'070.209 V
SI-Prefix	3	23.070 kV
Engineering	3	23.1e03 V
Scientific	3	2.307e04 V
None	3	2.31E+04 V

21 Formeleditor

TranAX bietet mit dem integrierten Formeleditor eine grosse Fülle von hilfreichen Funktionen zur Analyse der erfassten Kurven und deren weitere Aufbereitung. Berechnung von neuen Kurven und Messwerten können ohne grossen Aufwand durchgeführt werden.

Mit den über **100 mathematischen Funktionen** und Operatoren kann nahezu jede praxisorientierte Berechnung durchgeführt werden.
Eine komplette Liste mit Funktionen ist im Anhang gelistet.

Der Formel-Editor bieten den Komfort aktueller Programmierumgebungen: Syntaxhervorhebung (Syntax-Highlighting), automatische Vervollständigung, Anwenderspezifische Schriftgrösse, etc.
Neben dem Haupt-Formel-Textfeld können via "Datei/Funktion in separate Seite lade.." weitere Tabs mit andern Formel- oder Funktions-Dateien eingefügt werden. Dies dient vor allem dazu, bestehende Formeln oder Funktionen (nebst den Haupt-Formeln) zu inspizieren, vergleichen, ändern usw.

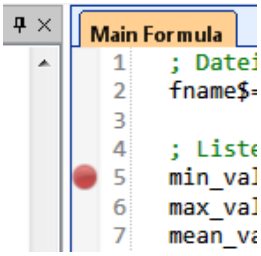


Der Formeleditor besteht aus den folgenden Komponenten:

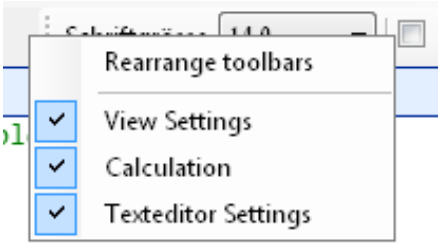
- Die **Berechnungen** bzw. **Formeln** werden im Textfeld in der Mitte **"Main Formula"** eingegeben. Jede Linie repräsentiert eine Funktion und jede Funktion wird in der Regel durch einen Zeilenumbruch getrennt.
- **Rechts** sind zwei Spalten für die **Resultate**, unten die Skalar-Werte, oben die berechneten Kurven. Die Kurven können dann mittels Drag & Drop in ein Kurvenanzeige-Fenster gezogen werden.
- Skalar-Werte unter "Results" können auch in Text-Feldern in Kurvenanzeigen zur Ansicht gebracht werden.
- Links oben befindet sich eine Liste mit allen **verfügbaren Operatoren und Kanälen**. Sie können mittels Doppelklick zum Textfeld hinzugefügt werden. **Unten** ist die Hilfe zu den einzelnen Funktionen eingeblendet.

21.1 Beschreibung der Toolbar

In der oberen Toolbar sind die folgenden Icons vorhanden:

	Mit der Schaltfläche **Berechnen** werden die Berechnungen manuell gestartet. Die Berechnung kann auch durch die Anweisung *"Calculate"* aus einer Autosequenz aufgerufen werden. Die Berechnung auch mit der Funktionstaste **F10** eingeleitet werden.
	Durch drücken des **Stop-Knopfs** während einer laufenden Berechnung wird diese abgebrochen.
	Wenn die Option **Automatisch berechnen** aktiviert ist, werden alle Berechnungen unmittelbar nach einer Messung durchgeführt.
	Speichern der Formel-Liste, es öffnet sich der entsprechende Dialog zur Eingabe des Dateinamens und zur Auswahl des Pfades. Die Datei-Erweiterung ".for" wird automatisch erzeugt.
	Öffnen-Dialog zur Auswahl einer Formel-Liste. Anschliessend wird diese im Textfeld "Main Formula" geladen.
	**Debugg Knöpfe**: Im Formelfenster kann links neben den Zeilennummern mit der linken Maustaste auf die graue Leiste geklickt werden, dadurch werden sogenannte Break-Points gesetzt. Mit dem grünen Pfeil "Start Debug" wird die Formel bis zum ersten Break-Point ausgeführt, die Berechnungen werden an dieser Stelle dann angehalten und die aktuellen Resultate können bei Traces und Results entnommen werden. "Step Over": Wie beim Knopf "Next Step" wird die Berechnung einer Zeile ausgeführt, wobei jedoch Unterfunktionen ("Functions") übersprungen werden. Durch Drücken des Knopfs mit dem Pfeil abwärts "Next Step" kann anschliessend jede weitere Zeile der Formel einzeln abgearbeitet werden. Mit dem Knopf "Step Out" werden die Berechnungen einer Schlaufe (for loop etc.) abgeschlossen. Es wird bei der nächsten Zeile nach der Schlaufe wieder gestoppt. Durch Klicken auf den Knopf "Next Breakpoint" können die Berechnungen weiter geführt werden.
	Schaltet das Fenster **"Function Browser"** zu. Aus diesem können alle Funktionen, Operatoren und Kanäle ausgewählt werden.
	Schaltet das Fenster **"Traces"** zu. Die in den Formeln berechneten Kurven werden in diesem aufgelistet. Solche Resultat-Kurven können via Drag & Drop in ein Kurven-Anzeigefenster gezogen werden.
	Schaltet das Fenster **"Results"** zu. Die in den Formeln berechneten Skalare (Zahlenwerte, keine Kurven) werden in diesem aufgelistet.

	Schaltet das Fenster **"Help"** zu. Hier wird zu jeder verfügbaren Funktion eine kurze Beschreibung angezeigt, sobald diese im "Function Browser" angewählt ist.
	Die **Schriftgrösse** im Fenster **"Formula"** lässt sich individuell einstellen und wird mit den Layout-Einstellungen gespeichert.
	Wenn diese Checkbox angewählt ist, werden während der Eingabe Vorschläge zur **automatischen Vervollständigung** eingeblendet. Mit den Tasten Pfeil auf- und abwärts kann der passende Vorschlag ausgewählt und mit der Taste "TAB" oder Leerschlag übernommen werden. Mit der Taste **"ESC"** wird die Vervollständigung abgebrochen.



Die einzelnen Felder können durch einen Rechtsklick auf der Toolbar ein-und ausgeblendet werden. Falls die Icons nicht mehr richtig ersichtlich oder angeordnet sind, können diese durch Klicken auf "Rearrange Toolbars" wieder geordnet werden.

21.2 Mit dem Formeleditor arbeiten

Zur Dokumentation von Formeln können Kommentare eingefügt werden. Diese sind durch einen Strich-Punkt ";" gekennzeichnet. Alles was nach einem Strich-Punkt auf einer Linie geschrieben ist, bleibt unberücksichtigt und wird als Kommentar interpretiert.
Zusätzlich kann in einem Kommentar ein Wort(teil) markiert und darauf ein Link zu einer Datei angefügt werden. Diese Datei wird dann (unsichtbar) in die Formel-Datei gespeichert. Sie kann eine ausführliche Beschreibung oder Anleitung zu den Formeln enthalten und kann wieder via Rechts-Klick über der entsprechenden Stelle geöffnet werden. Es können PDF-, Text- und Bilddateien angefügt werden.

Die einzelnen Berechnungen werden im Textfeld nach folgendem Muster eingegeben:

```
name[Einheit] = Ausdruck
```

name ist ein **Variablen-Name**, welcher das Resultat benennt. Das Resultat kann in nachfolgenden Formeln eingesetzt werden, um weitere Berechnungen damit durchzuführen. Die *Einheit* dient dazu, berechneten Kurven eine ***Einheit* zuzuweisen**. Diese muss aufgrund der Einheiten der Kurven, welche in die Berechnung einfliessen, sowie allfälliger Skalierungen bestimmt werden. Beispiel: C0A1 misst eine Spannung in Volt, C0A2 messe einen Strom in Ampere. Dann wäre folgende Berechnung korrekt:

```
Leistung[mW] = c0A1 * c0A2 * 1000
```

Der Variablen-Name *"name"* kann im Prinzip beliebig gewählt werden. Einzig sogenannte Schlüsselwörter (z.B. Cos, Sinus, If, For, to, as, Pi usw.) sind nicht erlaubt und erzeugen eine Syntax-Fehlermeldung.
Auf der **rechten Seite** des **Gleichheitszeichens** können **beliebige Ausdrücke** stehen, welche die **Grundoperationen** Addition und Subtraktion, Multiplikation und Division sowie Potenzfunktionen beinhalten. Das Vorzeichen-Minus wird ebenfalls akzeptiert (z.B. –4 oder –c0A1). Multiplikation und Division haben eine höhere Priorität als Addition und Subtraktion. Klammern kön-

nen beliebig gesetzt werden. Die **Operanden** können **Zahlen**, **Kanäle**, **Kurven** aus **TPC5- oder TDP-Dateien, Strings** (Zeichenketten)**, Arrays** oder vorher berechnete **Resultate** sein.
In der Regel werden die internen Berechnungen in doppelter Präzision (64Bit, 15 Dezimal-Stellen, Exponent > 300) durchgeführt. Gespeicherte Resultat-Kurven werden jedoch nur als 32Bit-Zahl abgelegt.

Kanäle werden mit einem "**c**", der **Geräte-Nummer** und der **Eingangsbezeichnung** angegeben (z.B.: C0A1, C0B3). Für Aufnahmen mit mehr als einem Block (Multiblock oder ECR) muss die **Blocknummer** angegeben werden (z.B. C0A1.2), andernfalls wird das Signal vom Block 0 berechnet. Mit einem **Apostroph (')** und einer Ziffer von 1 bis 2 kann ein Marker des Kanals ausgewählt werden:

```
Marker_1 = c0A1'1
Marker_2 = c0B3'2
```

Marker-Resultate werden durch die Analog-Werte 0 und 1 repräsentiert.

Kurven aus Dateien: Eine Datei wird mit deren Namen eingegeben:

```
File("dateiname.tpc5", 1)
```

Um die gewählte Datei nachträglich zu ersetzen, kann mit der rechten Maustaste auf den Dateinamen geklickt werden. Im Kontextmenü anschliessend "Set Filename.." klicken und mit dem Datei-Dialog die entsprechende neue Kurven-Datei auswählen.

Der **Index** repräsentiert die **Kurven-Nummer** in der Datei (0 entspricht der ersten Kurve, dieser Index ist nicht zu verwechseln mit den Kanal-Nummern!). Im oberen Beispiel wird also die zweite Signalkurve der Datei für die Berechnung verwendet.

Wenn über dem Index einer File-Funktion die rechte Maus-Taste gedrückt wird, werden alle Kurven in der Datei mit ihrem Namen der Reihe nach aufgelistet.

Sollen anstelle der Analog-Daten die Marker-Daten aus der Datei geholt werden, kann dies durch einen dritten Parameter angegeben werden:

```
File("Dateiname", Index).Block'Marker
```

Der Wert für den Parameter "Marker" muss entweder 1 oder 2 sein. Beispiel:

```
Signal = File("crash.tpc5",0)'1
```

Wenn kein Pfad mit dem Dateinamen angegeben wird, sucht das Programm die Kurve im Verzeichnis **DATA** des **aktuellen Experiments**. Für eine Datei in einem andern Verzeichnis innerhalb des aktuellen Experiments muss nicht die gesamte Pfadbezeichnung (z.B. "C:\User\USERNAME\Documents\TranAX\EXPERIMENT.exp\ref\name.TPC5") angegeben werden. Es genügt die Bezeichnung "..\ref\name.TPC5".

Was nach einem **Semikolon (;)** auf einer Zeile steht, wird vom Formel-Interpreter bis zum Zeilenende ignoriert. Auf diese Weise können erklärende **Kommentare** eingegeben werden.

21.3 Cursors platzieren

Die Funktion ***SetCrs**("Fenstername", "A", t)* **platziert die Cursor** in der Kurvenanzeige. Die resultierende Position der Cursor im Fenster bezieht sich auf die X-Achse und hängt von der eingestellten X-Achsen-Skalierung ab (relativ zum Start, relativ zum Triggerzeitpunkt und Samples). Diese Funktion ist nützlich, wenn mit Skalarfunktionen z.B. die Fläche unter einem Impuls bestimmt werden soll. Dazu kann der Puls-Anstieg und -Abfall mittels *TCross* gesucht werden, um danach die Cursor mittels *SetCrs("...","A")* und *SetCrs("...","B")* auf die **gefundenen Puls-Grenzwerte** zu setzen.

Weiteres Beispiel:
Die Start- und End-Zeit der Kurve bestimmen

```
t_start = TBegin(c0A1)
t_end = TEnd(c0A1)
```

Bei beiden Kurven den ersten Anstieg auf über 1V suchen

```
T1 = TCross(c0A1, t_start, t_end, 1)
T2 = TCross(c0A2, t_start, t_end, 1)
```

Die beiden Cursors bei den gefunden Durchgängen platzieren

```
SetCrs("Kurvenanzeige 1", "A", T1)
SetCrs("Kurvenanzeige 1", "B", T2)
```

21.4 String Variablen

String Variablen z.B. "Kurvenanzeige 1" oder "A" können mit den entsprechenden Funktionen deklariert werden. Beispiel:

```
value$ = "test.tpc5"
```

Das $-Zeichen in einer String-Variablen dient nur zur besseren Unterscheidung. Eine String-Variable kann auch ohne $-Zeichen vorgegeben werden.

Mehrere Strings können miteinander verknüpft (aneinander gereiht) werden:

```
name$ = "test"
extension$ = ".tpc"
version = 5
value$ = name$ + extension$ + version
```

Bei String-Operationen werden vorgegebene Zahlen automatisch als String interpretiert.
Zum Beispiel muss hier *version=5* nicht als *version="5"* geschrieben werden (die Funktion *version=2*2.5* würde das gleiche Ergebnis liefern).

21.5 Eigene Unterfunktionen definieren

Häufig verwendete Funktionsblöcke können in eigene Unterfunktionen ausgelagert werden. Dies dient der Vereinfachung und Übersichtlichkeit des Hauptprogramms.

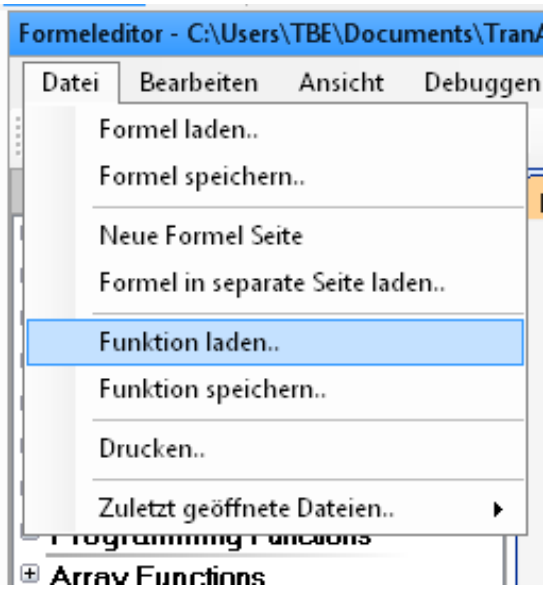
```
Function Name (parameters*)
 Function-BLOCK ; Berechnungen
 Name = value ; Rückgabewert
EndFunction
```

Der Rückgabewert wird als Formel mit dem Variablen-Namen, der dem Funktions-Namen entspricht bestimmt.

Es können auch mehrere Unterfunktionen in einer Datei definiert werden.
Beispiel mit einer Datei mit dem Namen "MathOperators.fnc":

```
Function Addition(value0, value1)
 Addition = value0 + value1
EndFunction
```

Normalerweise werden solche Unterfunktionen in eigenen Dateien abgelegt. Die Namen dieser Dateien haben die Erweiterung *.fnc.
Im Menü ***"Datei" / "Funktion laden"*** bzw. ***"Funktion speichern"*** können diese Funktions-Dateien verwaltet werden:



Um eine Unterfunktion verwenden zu können, muss die Datei mit der Unterfunktion oben in den Hauptformeln deklariert werden. Dies wird mit dem **Schlüsselwort "*using*"** gemacht:

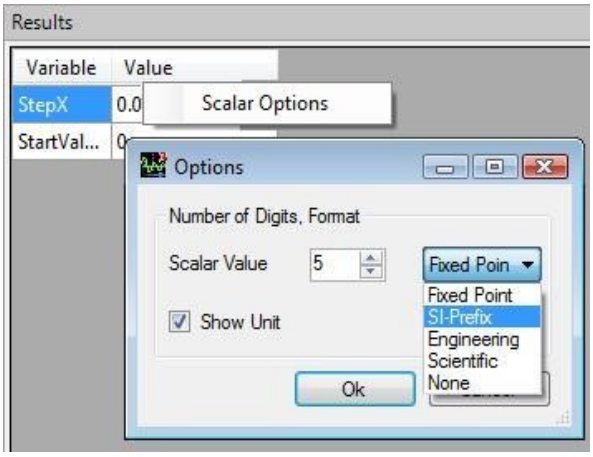
```
using "MathOperators.fnc"
result = Addition(2, 3)
```

Unterfunktionen können sich aber auch im gleichen Formel-Fenster unterhalb der Instruktion ***EndFormula*** befinden. Dies erleichtert vor allem die Arbeit bei der Entwicklung von eigenen Funktionen. In einem solchen Fall kann in den Haupt-Formeln die Anweisung ***using ()***" entfallen.
Unterfunktions-Aufrufe können sich auch in andern Unterfunktionen befinden.

Eine Sammlung von Standard-Unterfunktionen wird mit TranAX mitgeliefert.
Damit sind einer Erweiterung der Berechnungen keine Grenzen gestetzt.
Der Anwender hat auch die Möglichkeit eine eigene Sammlung von Funktionen anzulegen.

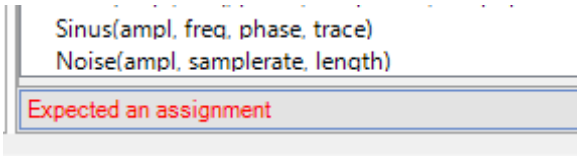
21.6 Zahlenformat der Skalar-Resultate

Die Darstellungsart der Skalar-Resultatwerte kann durch einen Rechtsklick auf die Zelle "Value" angepasst werden.



21.7 Fehlermeldungen

Allfällige Fehler in den Formeln, die durch den Parser festgestellt werden können, werden unten links im Formelfenster angezeigt. Mit einem **Doppelklick** auf einer allfälligen **Fehlermeldung** am unteren Ende des Formeleditors **springt der Text-Cursor an die fehlerhafte Stelle im Code**. Dies verkürzt die Suche nach der entsprechenden Fehlerquelle.



Oft liegt der Fehler in der unmittelbar davorliegenden Formel.

21.8 Übersicht der Funktions-Gruppen

Eine komplette Liste mit Funktionen ist im Anhang gelistet. Die verfügbaren mathematischen Funktionen sind in die folgenden Gruppen unterteilt:

Name	Beschreibung
Channels	Hardware Signal-Quellen
All Functions	Alle verfügbaren Funktionen in alphabetischer Reihenfolge
Base Functions	Basis-Funktionen
File Functions	Operationen die sich auf Dateien beziehen
Signal Analysis	In der Regel Einzelwerte als Resultat
Signal Processing	In der Regel Kurven als Resultat
Filter Functions	Diverse Filter Funktionen
Programming Functions	Wenn/Dann-, Schlaufen usw.
Array Functions	Funktionen zum Arbeiten mit Arrays
Exponential and Trigonometric	Trigonometrische Funktionen
Spectrum (FFT)	Erzeugen und Berechnen von Spektren
Report Generator	Bilden und Ausfüllen von Berichten
Recording Parameters	Vornehmen von Hardware-Einstellungen
Layout Waveforms	Eigenschaften von Kurven-Anzeigen
Auto Sequence	Steuerung mittels Makro-Befehlen
Signal Generations	Funktionen, die Kurven generieren
Misc. Functions	Div. weitere Funktionen

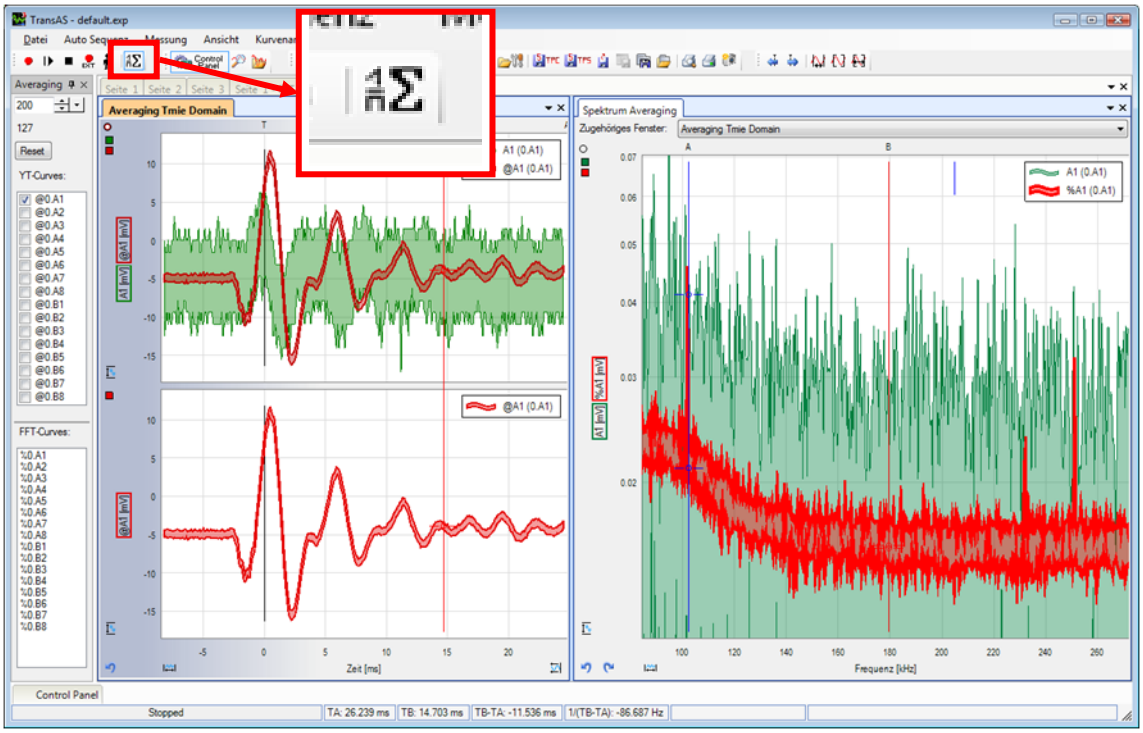
22 Mittelwertbildung über mehrere Aufnahmen

Die **Mittelwertbildung über mehrere Aufnahmen** ermöglicht, mehrere ähnliche, nacheinander aufgenommene Signale aufzusummieren und deren Mittelwert als Kurve darzustellen. Auf diese Weise können z.B. unregelmässige Störungen oder im Signal vorhandenes Rauschen vermindert werden.

Die Mittelwertbildung kann nur im Scope-Mode verwendet werden.

Um die Mittelwertbildung zu aktivieren, muss der Knopf **gedrückt sein**. Während die Mittelwertbildung aktiviert ist, ist auch das Averaging-Fenster geöffnet. Dieses wird mit Vorteil auf der linken Bildschirmseite angedockt.

22.1 Mittelwertbildung im Zeitbereich



Die untere Liste im Averaging-Fenster bezieht sich auf die Mittelwertbildung von Spektren.

Oben im Fenster ist ein Eingabefeld zu sehen, in welchem man die Anzahl der Aufnahmen eingibt, welche aufsummiert werden sollen. Die Zählanzeige darunter zeigt die Anzahl der bereits aufgenommenen (aufsummierten) Aufnahmen an. Wird die Limite erreicht, werden bei der nächsten Aufzeichnung die Summen gelöscht und eine neue Serie beginnt. Der Reset-Knopf dient dem vorzeitigen Zurücksetzen des Zählerstandes.

Durch Anklicken des Knopfs "Reset" wird vorerst nur intern festgehalten, dass bei der nächsten Aufnahme die Mittelwertbildung wieder bei eins beginnt. D.h. die aktuell vorliegenden gemittelten Kurven bleiben erhalten bis zum nächsten Aufnahme-Start.

In der Liste darunter sind alle Kanäle aufgelistet. Man kann individuell für jeden Kanal bestimmen, ob ein Mittelwert aufsummiert werden soll oder nicht.

Gleichzeitig dient diese Liste dazu, mittels Drag & Drop die gemittelten Kurven zur Darstellung zu bringen. Man kann mit der Maus einen Kanal ziehen und in einem Anzeige-Fenster fallen

lassen. Um die gemittelten Kanäle von den "normalen" Kanälen, d.h. von der zuletzt gemachten aktuellen Aufnahme unterscheiden zu können, haben die gemittelten Kanäle ein @ in ihrer Bezeichnung.

Nach jeder Aufnahme werden die zur Mittelwertbildung aktivierten Kanäle ausgelesen und zur aktuellen Summe hinzu addiert. Dann werden alle Kurvenanzeigen neu dargestellt. So sieht man immer den aktuellen Mittelwert aus den bereits gemachten Aufnahmen. Wenn die Anzahl der Messungen die vorgegebene Limite erreicht, werden die Summen gelöscht und eine neue Serie beginnt.
Wenn im Control-Panel nicht "Single Shot" eingestellt ist (= Auto-Start), werden kontinuierlich Messungen gemacht und aufsummiert, bis die Limite erreicht ist. Dann wird angehalten. Durch Drücken des Start-Knopfs kann eine neue Serie gestartet werden.

Die aktuellen Mittelwert-Kurven können auch im Formel-Editor zu Berechnungen herangezogen werden, indem man sie mit **@** kennzeichnet, z.B. @c0A1 (anstatt c0A1) für Kanal 1.

Beispiel:

```
Save xx.tpc5, @0A1, @0A3-4
```

Um gemittelte Kurven in der Auto Sequenz zu speichern, muss die Installierte Software eventuell aktualisiert werden. Die folgenden Versionen werden als **Minimum vorausgesetzt**:

TranAX: 3.2.1.702 (Menü ***"Hilfe" / "Info"***)

22.2 Mittelwertbildung von Spektren

Auf die gleiche Art wie bei der Mittelwertbildung im Zeitbereich lassen sich durch **FFT ermittelte Spektren** aufsummieren und als Mittelwert darstellen, wobei nur von den Amplituden-Werten eine Mittelwertbildung durchgeführt wird. Die Phasenwerte werden nicht gemittelt.
Auch hier muss mit dem Knopf die Mittelwertbildung aktiviert sein.
Im unteren Teil des Average-Fensters sind alle Kanäle aufgelistet, welche für ein Mittelwert-Spektrum zur Verfügung stehen.
Ein "**%**" vor der Kanal-Nummer kennzeichnet, dass es sich um ein gemitteltes Spektrum handelt.
Es wird nur von denjenigen Signalen ein Mittelwert-Spektrum erzeugt, welche via Drag & Drop aus dieser Liste in ein geöffnetes Spektrum-Fenster gezogen wurden.

Aus dem Spektrum-Bild oben geht schön hervor, wie durch Mittelung von Spektren Frequenzen zum Vorschein kommen, welche sonst im Rauschen untergehen würden.

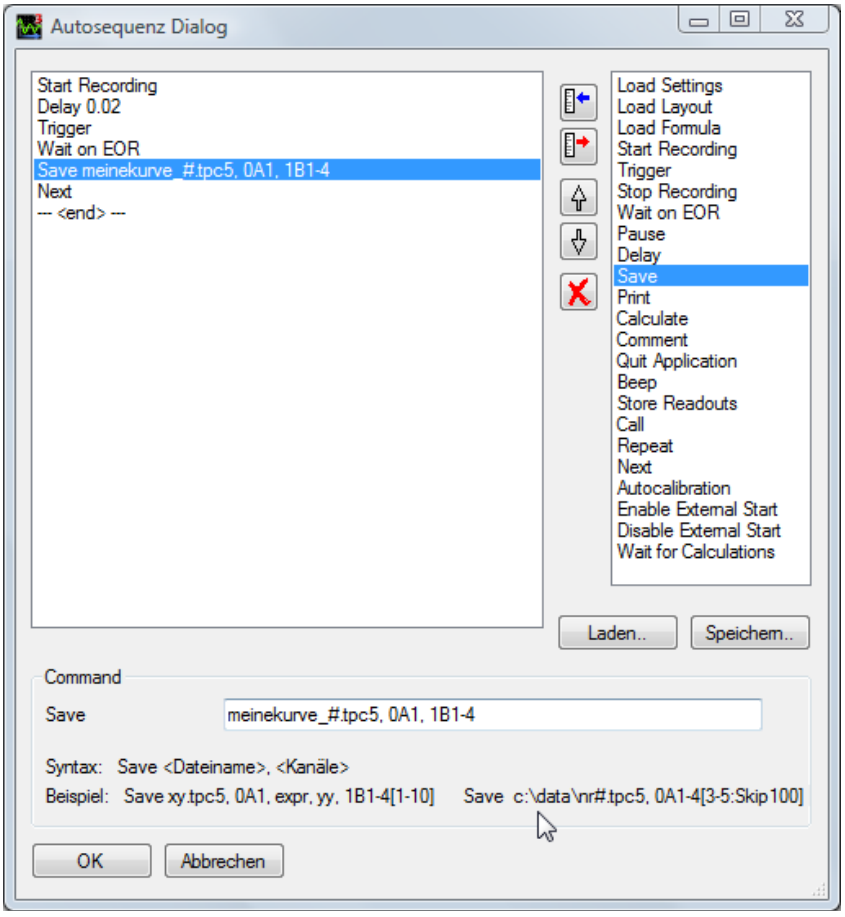
Falls bei Spektrum-Mittelung gewisse Spektrum-Parameter (z.B. FFT-Gewichtung oder Verstellen der Zeitmarken im zugehörigen Y-T-Fenster) geändert werden, erfolgt ein automatisches Zurücksetzen des Zählers.

Eine Mittelwertbildung braucht Rechenzeit und Speicherplatz, deshalb sollte man nur die benötigten Kanäle aktivieren. Auch sollte die Blocklänge möglichst kurz gewählt werden (vor allem beim Mitteln von Spektren).

23 Auto Sequenz

Mit Hilfe einer Autosequenz können einfache, sich **wiederholende Messabläufe automatisiert** werden. Eine **Autosequenz** besteht aus einer **Liste von Befehlen**, welche der Reihe nach abgearbeitet werden.

Eine komplette Liste der Auto Sequenz Befehle ist im Anhang gelistet.



In der **rechten Tabelle** sind alle **verfügbaren Befehle** aufgelistet. In **der linken Tabelle** kann die **Autosequenz zusammengestellt** werden.

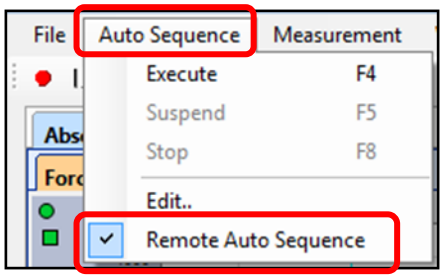
	Fügt den markierten Befehl aus der rechten Tabelle in die Autosequenz ein.
	Entfernt den markierten Befehl aus der Autosequenz.
	Verschiebt den markierten Befehl in der Autosequenz eine Position nach oben.
	Verschiebt den markierten Befehl in der Autosequenz eine Position nach unten.
	Löscht die gesamte Autosequenz.

Befehlsabfolge einer Autosequenz können in Dateien gespeichert und mit den entsprechend beschrifteten Knöpfen wieder geladen werden.

Gewisse Befehle verlangen noch einen oder mehrere zusätzliche Parameter. Diese müssen im **Command Feld unterhalb der Befehls-Liste** angegeben. Die Zeile unter dem Command-Eingabefeld erklärt immer die Syntax des aktuell markierten Befehls.

23.1 Auto Sequenzen in autonom laufenden TraNET-Geräten

Geräte, die via Ethernet gesteuert werden (z.B. TraNET-FE Geräte), können mit eigenen Autosequenzen (Makros) geladen werden. Diese laufen dann in den Geräten autonom ab. Dadurch lassen sich Messreihen aufzeichnen, auch wenn kein Bedien-PC mehr angeschlossen ist.

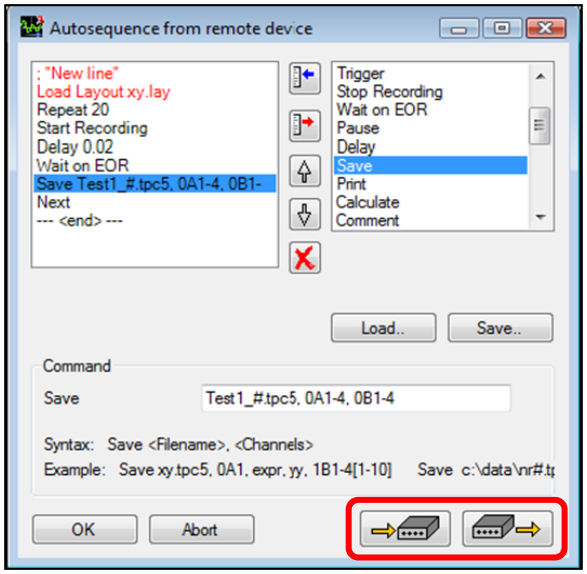


Wenn im Menu Auto Sequenz "Remote Auto Sequence" markiert ist, bezieht sich alles (Ausführen, Bearbeiten usw.) auf die autonomen Makros in dem verbundenen Gerät.

Befehle, welche in den autonomen Makros nicht ausgeführt werden können (z.B. Calculate, Print, Laden von Layouts usw.) sind in roter Schrift dargestellt. Sie stören nicht und werden beim Ablaufen einfach ignoriert.

Das "Save"-Kommando unterstützt nur tpc5-Dateien, ohne zusätzliche Parameter, wie Block-Bereich, Data-Reduktion usw.

Bei einem Wechsel zu einem anderen Experiment, wird eine laufende Autosequenz ohne Warnung abgebrochen.



Autosequenz-Makros können in beiden Fällen (remote oder direkt) als Datei auf dem Bedien-PC geladen und/oder gespeichert werden.

Mit den beiden Knöpfen lassen sich erstellte Makros in das angeschlossene Gerät Laden oder zurück Lesen.

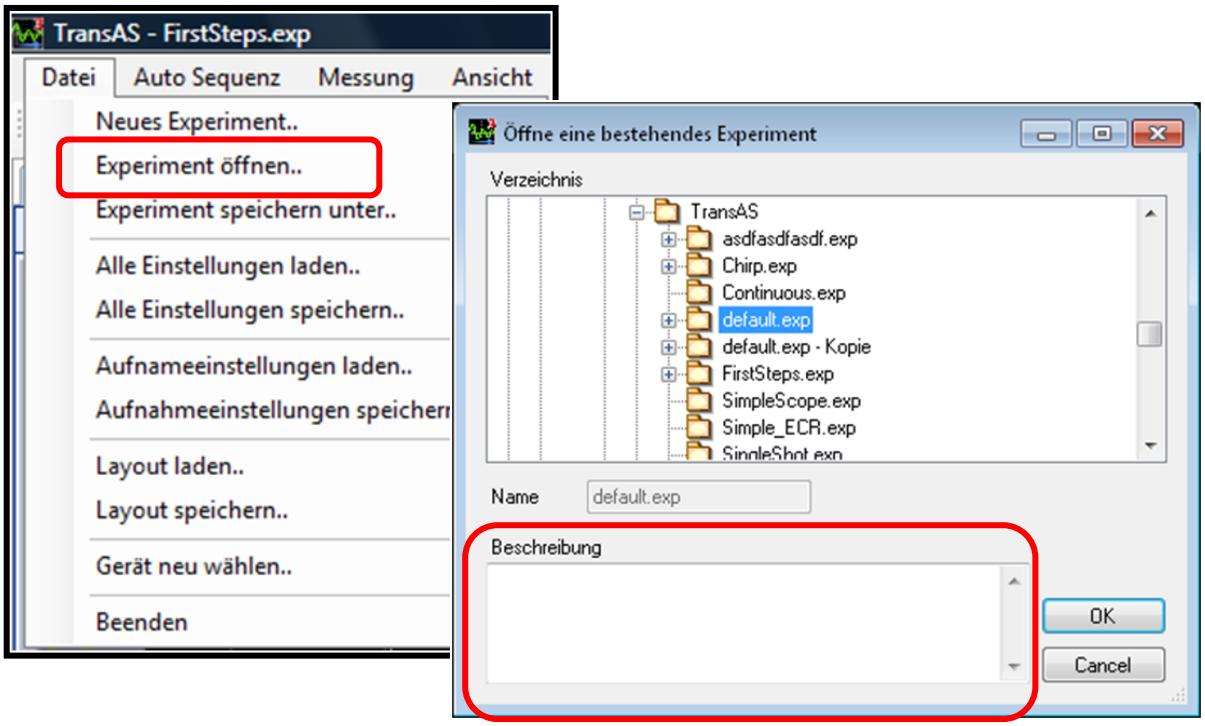
Um die Möglichkeit der autonomen Autosequenz zu verwenden, muss die Installierte Software eventuell aktualisiert werden. Die folgenden Versionen werden als **Minimum vorausgesetzt**:

TranAX:	**3.3.2.922**	(Menü ***"Hilfe" / "Info"***)
TPC-Server:	**1.4.2**	(Control Panel /)
TraNET FE:	**1.4.2**	(TraNetConfiguration.exe / Show Logfile / Server)

Ein Firmware Update wird ebenfalls mit TraNetConfiguration.exe durchgeführt.

24 Experimente

In TranAX bedeutet der Begriff "**Experiment**" eigentlich ein **Mess-Projekt**. Darin werden Einstellungen wie Messbereich, Abtastrate, Kanalnamen, die Anordnung der Fenster, Formeln, Autosequenzen usw. für einen bestimmten Messvorgang gespeichert. Des Weiteren kann für jedes Experiment eine kurze **Beschreibung** eingegeben werden.



Für **jedes Experiment** wird ein **Unterverzeichnis** XYZ**.EXP** angelegt, wobei XYZ der Name des Experimentes ist. Gleichzeitig werden innerhalb dieses Experimentes auch die **Unterverzeichnisse DATA** und gegebenenfalls **EXPR** erstellt.

Sobald mit dem Formeleditor eine neue Kurve berechnet wurde, wird diese im Ordner EXPR als .tpc5 Datei abgelegt. Auch dieses Unterverzeichnis wird durch TranAX automatisch erstellt.

Bei **"Experiment speichern unter.."** werden alle **momentanen Einstellungen** der Datenerfassungs-Hardware (Control-Panel-Settings, Lastset.tps.xml) sowie die Anordnung und Grösse der Fenster (Layout, Lastlay.LAY), die Autosequenz (Lastauto.AUT) und die Formeln (Lastfor.FOR) **abgespeichert**.

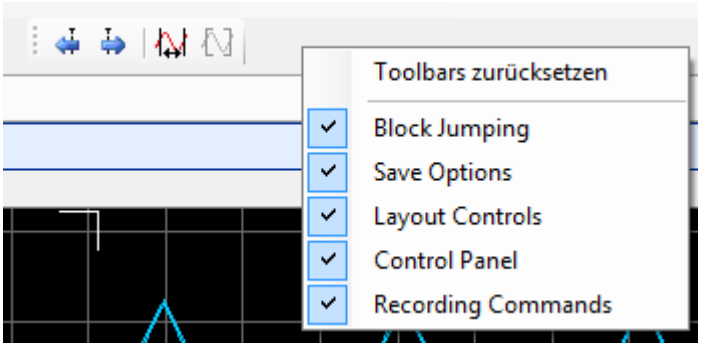
Beim **Laden** eines **Experimentes** wird alles wieder gemäss den ***Last.****-**Dateien** eingestellt. So kann auf einfache Weise zwischen verschiedenen Messkonfigurationen umgeschaltet werden. Beim **regulären Beenden** von TranAX werden im aktuellen Experiment alle ***Last.****- **Dateien erzeugt** und der Name des Experimentes für den nächsten Programmstart registriert.

25 Weitere Menüpunkte

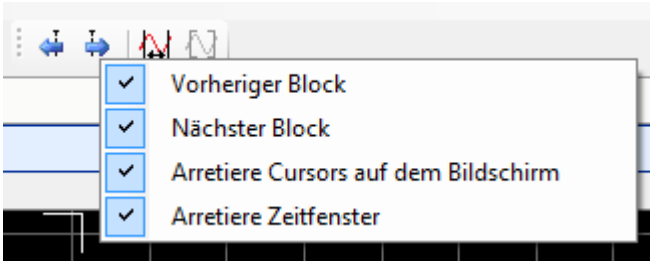
Alle verfügbaren Shortcuts (Tastatur-Kurzbefehle) in TranAX können auf einen Blick angezeigt werden. Dazu öffnet man die Liste über das Menü "Hilfe" / "Shortcut Übersicht" oder drückt die **F2 Taste**.

25.1 Benutzerspezifische Toolbars

Falls man die Toolbars benutzerspezifisch anpassen möchte, klickt man mit der rechten Maustaste auf die jeweilige Toolbar-Gruppe:

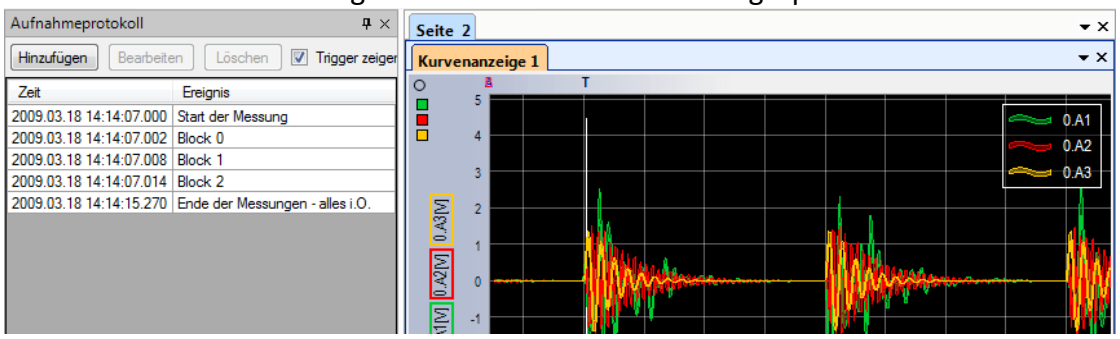


Die verschiedenen Pop-up-Menüs ermöglichen es, Toolbars ein- oder auszuschalten, Toolbars zurückzusetzen welche überlappend erscheinen oder die einzelnen Control Buttons ein- oder auszuschalten:



25.2 Aufnahmeprotokoll (Menü "Ansicht")

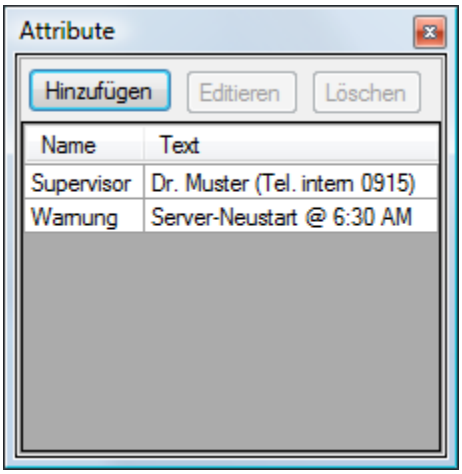
Das Aufnahmeprotokoll ist geeignet, um jederzeit Bemerkungen zu den Kurven-Daten während einer Aufzeichnung im Kontinuierlichen- oder ECR-Mode hinzuzufügen. Diese werden automatisch mit einem Zeitstempel versehen und können auf dem Bildschirm am entsprechenden Ort auf der Zeitachse angezeigt werden. Sie können auch im Nachhinein ergänzt oder editiert werden und werden mit den Signalkurven in eine TPC5-Datei gespeichert.



Falls Sie **Trigger zeigen** aktivieren, werden alle Trigger-Ereignisse inkl. Zeitstempel und entsprechende Kanäle aufgelistet. Sie können auch Ihre eigenen Kommentare mit der Schaltfläche **Hinzufügen** eingeben.

Klicken Sie im **Menü *"Anzeige" / "Aufnahmeprotokoll"*** um diese anzuzeigen.

25.3 Attribute (Menü "Ansicht")

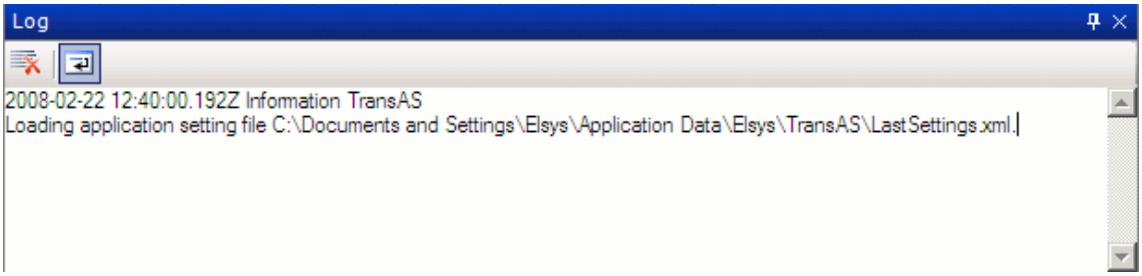


Unter ***"Attribute"*** können, im gleichen Sinn wie unter Aufnahmeprotokoll, **Text-Einträge** gemacht werden. Diese werden dann in den TPC5-Kurvendateien gespeichert. Attribut-Einträge haben jedoch im Unterschied zu Aufnahmeprotokollen keinen Zeitstempel. Attribute können Einträge wie Projektbezeichnung, Versuchsteilnehmer oder Hinweise zum Messaufbau enthalten.

Klicken Sie im **Menü *"Anzeige" / "Attribute"*** um diese anzuzeigen.

25.4 Fehlerprotokoll (Menü "Ansicht")

Das Fenster **Fehlerprotokoll**, vom Menü ***"Ansicht" / "Fehlerprotokoll"*** aus aufgerufen, zeigt alle aufgetretenen Fehler. Die Fehler-Liste kann nützliche Hinweise bieten, wenn sich TranAX fehlerhaft verhält.


Log
2008-02-22 12:40:00.192Z Information TransAS
Loading application setting file C:\Documents and Settings\Elsys\Application Data\Elsys\TransAS\LastSettings.xml.

25.5 Hardware Settings Viewer (Menü "Ansicht")

Via Menü "Ansicht / Hardware Settings Viewer" öffnet sich ein Fenster, das dem Control-Panel entspricht. Darin kann dann eine bestehende Setting-Datei (*.tps.xml), welche vorab gespeicherte Aufnahmeeinstellungen enthält, geladen werden.

Dadurch lässt sich im Nachhinein (auch wenn keine Hardware angeschlossen ist) nachvollziehen, welche Einstellungen bei den entsprechenden Messungen vorlagen (natürlich mussten dazu die Aufnahmeeinstellungen in einer Datei gespeichert worden sein).

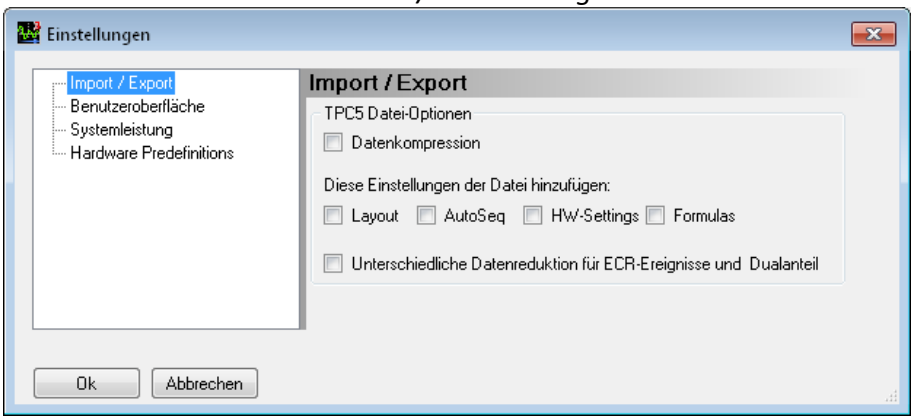
Auch hier kann via <Strg>-S eine Text-Datei erzeugt werden, welche die Einstellungen der in der Liste markierten Kanäle enthält.

25.6 Sprache (Menü "Extras")

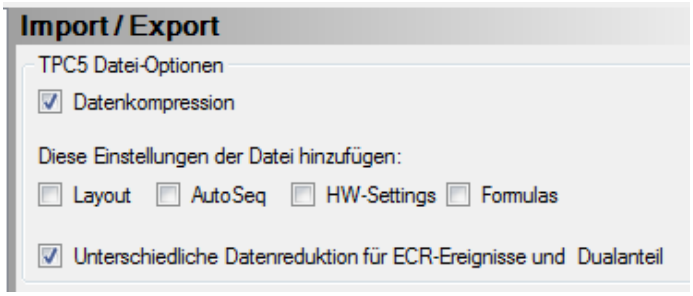
Die Sprache kann direkt im Menü "*Extras" / "Language"* geändert werden. Es stehen die beiden Sprachen Deutsch oder Englisch zur Auswahl.
Anschliessend öffnet sich ein Dialog, welcher zum Neustart von TranAX auffordert, um die Änderungen zu übernehmen.

25.7 Einstellungen (Menü "Extras")

Mit der Auswahl Menü "*Extras" / "Einstellungen*" öffnet sich das Fenster **Einstellungen**.



25.7.1 Import/Export (Menü "Extras")



Die hier vorgenommenen **Einstellungen** werden als **Standardwerte** in den Anderen Dialogen übernommen. Die Option **Datenkompression** wird anschliessend im **Speichern Dialog** als Standardeinstellung wieder angegeben.

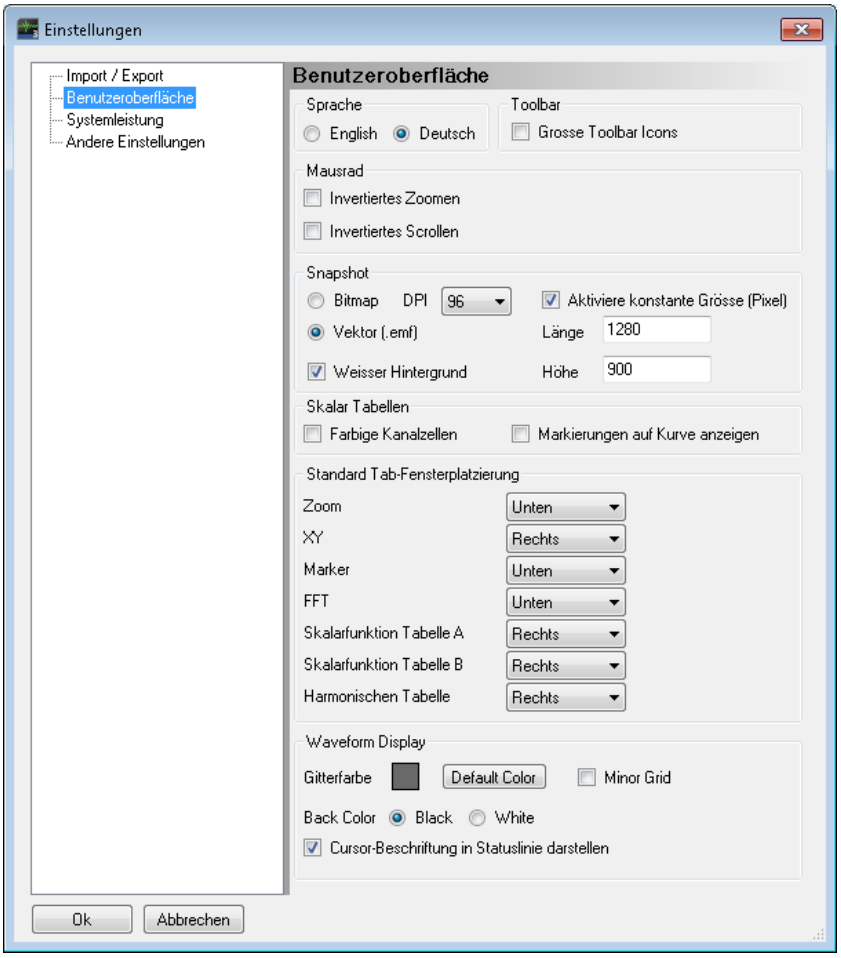
Bei den TPC5 Datei-Optionen können die folgenden Einstellungen vorgenommen werden:

Datenkompression	Diese Einstellung bestimmt die Grundeinstellung des Fensters **Speichern.**
Einstellungen der Datei hinzufügen	Das Datenformat TCP5 bietet die Möglichkeit neben den Kurven-Daten auch Einstellungen in derselben Datei mit abzuspeichern
Layout	enthält alle Informationen der Bedieneroberfläche
Auto Seq	enthält die Informationen der Auto Sequenz
HW-Settings	enthält die Hardware-Einstellungen aus dem Control Panel
Formulas	enthält die Informationen des Formel Editors
Unterschiedliche Datenreduktion für ECR-Ereignisse und Dualanteil	Falls im ECR Datenerfassungs-Mode der Dual-Anteil mit aufgezeichnet wurde, kann hier freigeschaltet werden, dass ECR- und Dual-Anteil unterschiedlich reduziert werden können

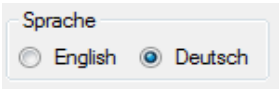
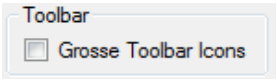
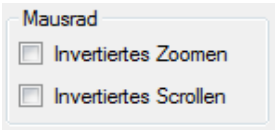
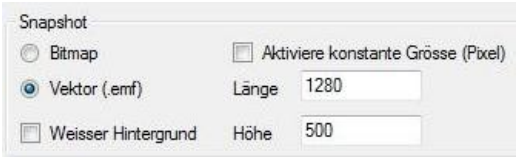
Im ECR Datenerfassungs-Mode kann der langsamere Dual-Anteil mit aufgezeichnet werden. Das Eingabefeld **Reduktion für ECR-Events** ist nur sichtbar, falls unter dem Menü ***"Extras" / "Einstellungen" / "Import /Export"*** die Einstellung "**Unterschiedliche Datenreduktion …**" aktiviert ist.

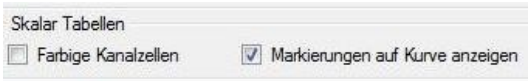
Für weitere Informationen zum Speichern von Signalkurven in das TPC5 Format, siehe Abschnitt Speichern.

25.7.2 Benutzeroberfläche (Menü "Extras")



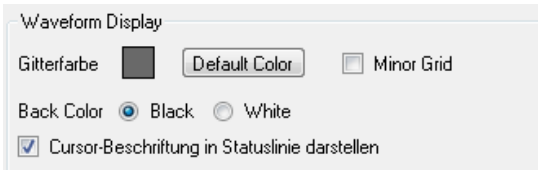
Hier könne Standardeinstellungen zur Benutzeroberfläche vorgenommen und definiert werden.

	Wie im Menü ***"Extras" / "Sprache"*** kann auch hier die Sprache festgelegt werden. *Nach einem Wechsel ist ein Neustart von TranAX notwendig um die Änderungen zu übernehmen.*
	Die Icons für die Aufnahme-Kommandos können auch im Grossformat angezeigt werden
	Die Richtung, in der eine Kurve verschoben oder ein- und ausgezoomt wird kann mit Aktivieren des Invertierens umgekehrt werden.
	Die Kurvenanzeige wird mit Klick auf das Kamera-Icon in die Zwischenablage gelegt. Falls **Vektor** angewählt ist, so wird mit dem Bild der Kurvenanzeige eine Vektorgrafik im Enhanced Metafile-Format angelegt. Zudem kann die Grösse in Pixels, wie der Bildausschnitt danach erscheinen soll, gewählt werden. Wenn "***Weisser Hintergrund***" markiert ist, wird der Hintergrund weiss gesetzt. Achtung: Helle Kurvenfarben sind dann z.T. schlecht sichtbar auf dem Bild. Daher sollten die Kurvenfarben eher dunkel gewählt werden.

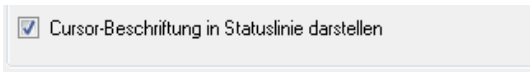


Wenn "***Farbige Kanalzellen***" eingeschaltet ist, wird die Hintergrundfarbe von Resultat-Zellen in Skalar-Tabellen auf die Farbe der jeweiligen Signalkurve gesetzt.
Wenn "***Markierungen auf Kurven..*** " eingeschaltet ist, werden Kurven nach entsprechenden Skalar-Berechnungen im zugehörigen Kurvenfenster mit Symbolen versehen. Es sind dies kleine Kreise bei Amplituden-Werten (z.B. Maximum, Peak-Peak usw.) oder kleine Rechtecke bei Zeit-Angaben (z.B. RMS-Periodic usw.).

Definiert die Position an der ein neues Fenster angelegt wird.

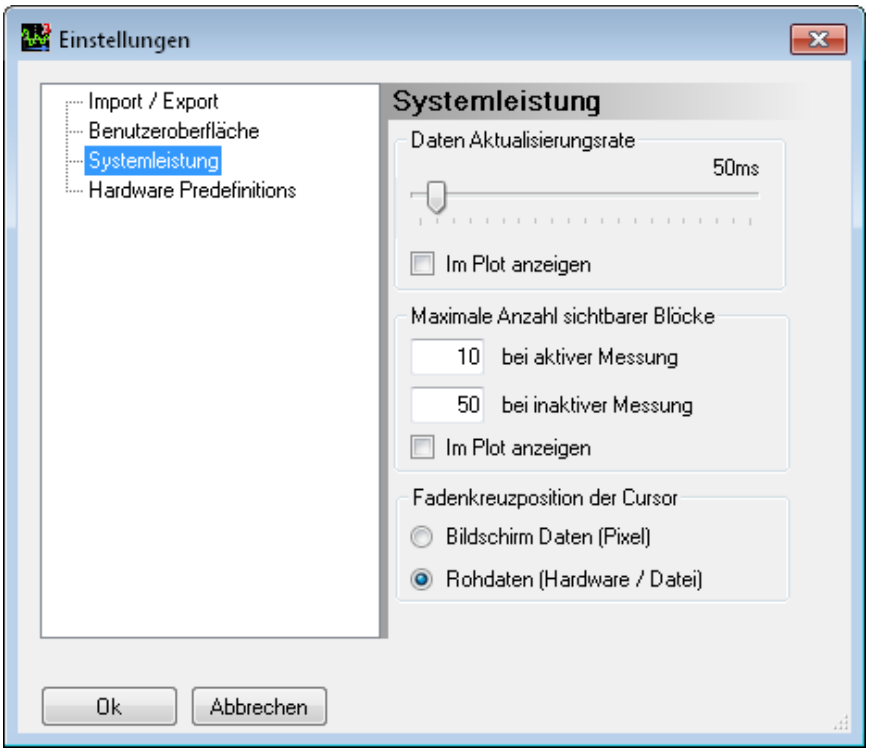


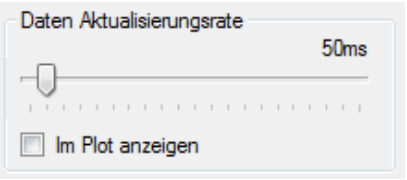
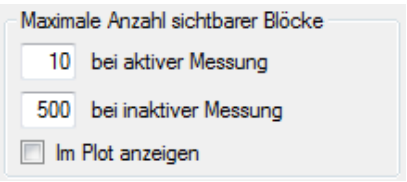
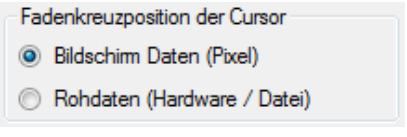
Via diesen Parameter kann die Gitter- und Hintergrundfarbe festgelegt werden.
Mit "Minor Grid" kann auch das feinere Gitter zugeschaltet werden.



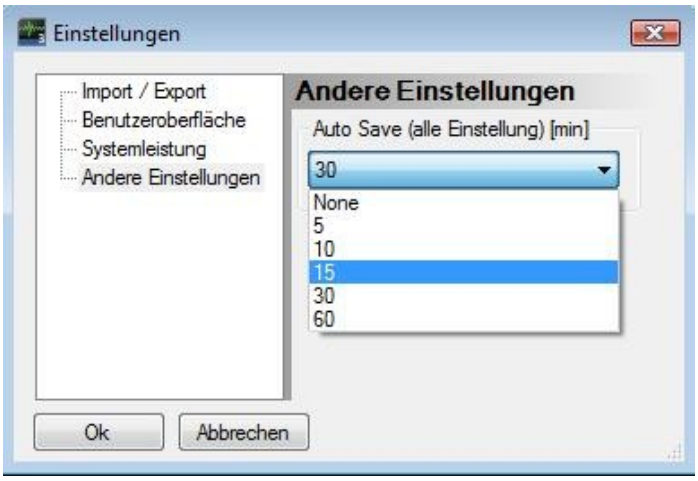
Die Cursor Buchstaben (A, B, C …) können oben innerhalb der Kurvenanzeige oder auf der Statuslinie darüber dargestellt werden.

25.7.3 Systemleistung (Menü "Extras")



	Hier wird angegeben, in welchen Intervallen die Daten der Kurvenanzeige aktualisiert werden sollen. TranAX verwendet diese Information um zu gewichten, wie die Systemleistung zwischen Datenvisualisierung und Datenauswertung erfolgen soll. Es gilt herauszufinden, welcher Wert für welche Anwendung geeignet ist. Wenn TranAX z.B. im kontinuierlichen Mode mit vielen Kanälen nur noch stark verzögert reagiert, sollte die **Daten Aktualisierungsrate** höher gestellt werden. Mit dem Knopf **Im Plot anzeigen** wird gewählt, ob diese Information in der Kurvenanzeige eingeblendet werden soll oder nicht.
	Mit dieser Einstellung kann festgelegt werden, wie viele Datenblocks bei aktiver oder inaktiver Messung maximal dargestellt werden sollen. Diese Einstellung bezieht sich auf Daten, die im Block- oder ECR-Mode erfasst wurden. Mit dem Knopf **Im Plot anzeigen** wird gewählt, ob diese Information in der Kurvenanzeige eingeblendet werden soll oder nicht.
	Die Y-Cursor-Position wird entweder aus den vorverarbeiteten Daten entnommen, die für die Bildaufbereitung verwendet werden, oder aus den Rohdaten. Die Selektierung *Rohdaten* zwingt TranAX, neue Daten anzufordern und kann deshalb die Systemleistung beeinträchtigen.

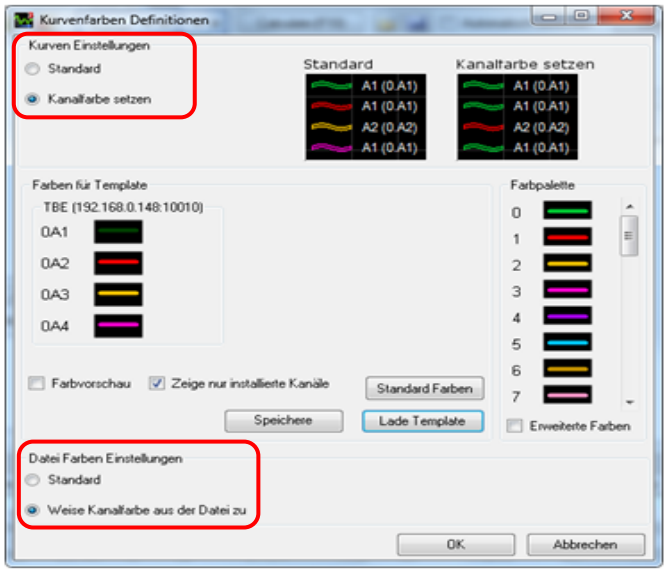
25.7.4 Andere Einstellungen: Auto Save (Menü "Extras")



Automatische Speicherung der Einstellungen.
Bei einem ungewollten Beenden von TranAX (Stromausfall usw.) werden eventuell die Einstellungen nicht in den Last.*-Dateien gespeichert. Alle vorgenommenen Änderungen seit dem letzten gewollten Speichern wären dabei verloren.
Unter "Extras / Einstellungen / Andere Einstellungen / Auto-Save" kann ein Zeitintervall vorgegeben werden, wonach die Einstellungen automatisch gespeichert werden.
Die dabei erzeugten Dateien befinden sich im Verzeichnis "AutoSavedSettings" des aktuellen Experiment-Verzeichnisses. Die Namen der Dateien sind nummeriert. Sie können via "Datei / Alle Einstellungen laden.." wieder aus dem Verzeichnis "AutoSavedSettings" aktiviert werden.

25.8 Kurven-Farben einem Kanal zuweisen (Menü "Extras")

Es gibt die Möglichkeit, den **einzelnen Kanälen** eine Farbe zuzuweisen, welche dann immer für denselben Kanal verwendet wird. So kann man für sein spezifisches Experiment die Farben den Kanälen zuteilen.



Im Menü unter *"Extras" / "Kanalfarben Einstellungen"* hat man einen Dialog zur Verfügung, welcher dies ermöglicht.

Damit die **Farben benutzerspezifisch** zugeordnet werden können muss unter **"Kurven Einstellungen" "Kanalfarbe setzen" aktiviert** sein.
In der Gruppenbox **"Farben für Template"** hat man nun die Möglichkeit, **jedem Kanal** eine **Farbe zuzuweisen**. Die Farbe kann entweder über den "Color Chooser" (Farbe anklicken) oder via **Drag & Drop** aus der **Farbpalette rechts** gewählt werden. Bei der Farbpalette kann man die abgestuften Farben abrufen indem man "Erweiterte Farben“ aktiviert.

Hat man einmal seine Farbauswahl getroffen, kann man diese via **"Farbvorschau"** direkt in TranAX betrachten und beurteilen. Möchte man ein **Template** vorbereiten, welches die Farbzuweisung nicht bloss zu den Kanälen der installierten Module zulässt, kann man **"Zeige nur installierte Kanäle" deaktivieren**. Danach können die Farben der Kanäle von 8 Devices mit je 8 Modulen zu je 8 Kanälen definiert werden.

Man kann die getroffene Farbwahl als **Template** mit benutzerspezifischem Namen **speichern**. Mit "OK" werden die gewählten Farben definitiv zugewiesen und die Maske geschlossen. Beim Verlassen des aktuellen Experiments oder beim Schliessen von TranAX wird automatisch die

Datei **"Lastcolors.ctf"** angelegt. Beim Erzeugen eines **neuen Experiments** wird automatisch die **aktuelle Farbauswahl übernommen**.

Einstellungen zu Kurvenfarben:

Standard	Jede Kurve erhält eine Farbe, welche von TranAX aufgrund der **Reihenfolge in den Kurvenfenstern automatisch** gewählt wird.
Setze Kanalfarbe	Jeder Kanal erhält die **selbstdefinierte Farbe.**

Datei Farben Einstellungen:

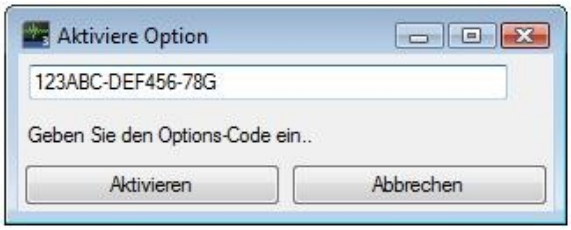
Standard	Jede Kurve erhält eine Farbe, welche von TranAX aufgrund der **Reihenfolge in den Kurvenfenstern automatisch** gewählt wird.
Weise Kanalfarbe aus der Datei zu	Kurven aus einer TPC5-Datei erhalten die Farben die **ursprünglich beim Speichern gewählt waren**.

25.9 Software Optionen Installieren (Menü "Extras")

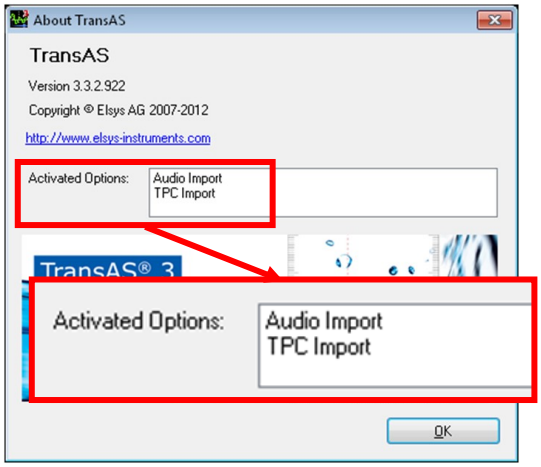
In TranAX sind zurzeit folgende Optionen verfügbar:

Name	Beschreibung
Import	Nachdem eine Import-Option installiert wurde, lässt sich im Menu "Datei/Imoprtieren..." eine der nachfolgenden Datei-Typen importieren. • *.TPC (aus TransAS 2) • *.ASD (ASCII aus TransAS 2 • *.MP3, *.WAV (Audio-Dateien) Des Weiteren können mit Hilfe eines sog. Wizards alle Arten von ASCII (Text) Dateien importiert werden. Beim Importieren werden die Fremddateien in das *.TPC5-Format kopiert und können danach in TranAX als solche verwendet werden.
ActiveX	Aktiviert die ActiveX Schnittstelle, dadurch kann TranAX von einem andern Programm gesteuert werden.
Report-Generator	Diese Option erlaubt die Erzeugung von EXCEL-Berichten. Nachdem sie installiert wurde, kann im Formel-Editor auf die entsprechenden Funktionen zugegriffen werden.

Bevor eine Option verwendet werden kann, muss sie **installiert (aktiviert)** werden. Die Installation erfolgt im Menü ***"Extras" / "Optionen installieren"***.



Hier kann die Serie-Nummer (Optionscode) eingegeben werden. Mit einem Klick auf **"*Aktivieren*"** kann der Code (Serie-Nr.) überprüft und aktiviert werden. Nach Eingabe einer gültigen Serie-Nummer kann das Fenster mit **"*Abbrechen*"** geschlossen werden.



Unter ***"Hilfe" / "Info"*** sind die aktivierten Optionen aufgelistet.

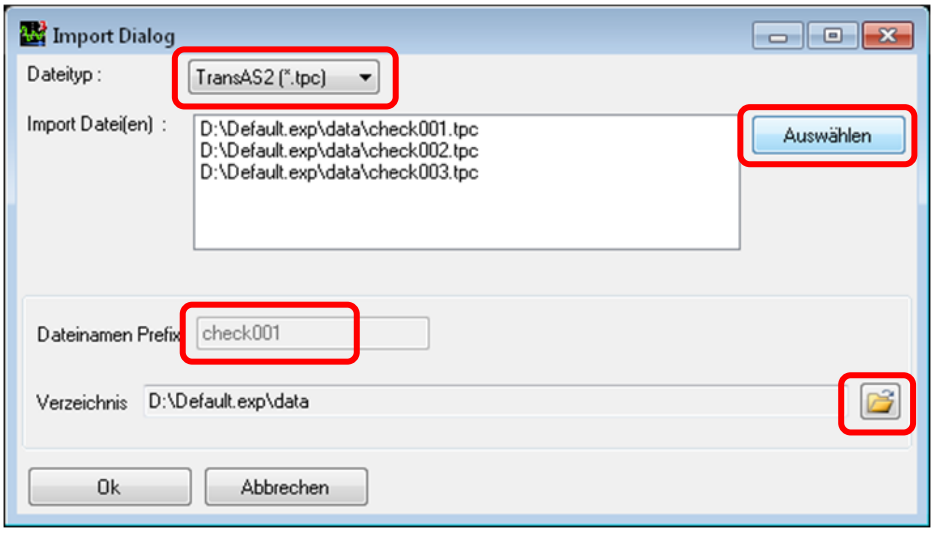
Die Serie-Nummer finden Sie im Deckel Ihres Handbuchs. Falls Sie eine Option nachträglich erwerben möchten, wenden Sie sich bitte an die für Sie zuständige Vertretung.

25.10 Import-Option

Verschiedene Datei-Formate lassen sich in TranAX importieren. Dabei werden diese Dateien ins TPC5-Format konvertiert (kopiert), sodass sie danach in TranAX analysiert werden können. Folgende Formate können importiert werden:

- TPC-Dateien (aus TranAX-Version 2)
- ASCII-Dateien (ASD-Dateien, auch aus TranAX-Version 2)
- Allgemeine ASCII-Dateien (Importieren via Wizard)
- Audio-Dateien

Via Menü ***"Datei" / "Importieren" / "Import Dialog"*** kann die Import-Maske geöffnet werden. Falls dieser Menü-Eintrag nicht vorhanden ist, muss die Option zuerst aktiviert werden.



Mit dem Parameter ***"Dateityp"*** kann das Datei-Format (TPC, ASCII, Audio etc.) gewählt werden. Danach muss via ***"Auswählen"*** eine oder mehrere Original-Dateien selektiert werden. Eventuell muss man dabei zuerst in das Verzeichnis der Original-Dateien navigieren.
Falls nur eine einzige Datei gewählt wurde, kann mit dem Parameter ***"Dateinamen .."*** ein (ev. neuer) Name gewählt werden. Ansonsten werden die Namen der Original-Dateien übernommen. Die Namenserweiterung "TPC5" wird automatisch angehängt.
Unter ***"Verzeichnis"*** muss der Zielort der erzeugten TPC5-Dateien eingestellt werden. Dies ist in der Regel der "data"-Ordner des aktuellen Experiments.
Eine bereits existierende TPC5-Datei wird ohne Warnung überschrieben.

Die konvertierten TPC5-Dateien werden automatisch im Signalquellen Browser hinzugefügt. Dieser lässt sich via Menü ***"Ansicht" / "Signalquellen Browser"*** öffnen. Via Signalquellen-Browser können ja dann die Kurven in die Anzeige-Fenster gezogen werden.

Anhang

26 Funktionsgruppen im Formeleditor

26.1 Gruppe "Channels"

GetChannel (device, board, input, blockNr)	Gibt die Signal-Kurve aus dem entsprechenden Eingang zurück. Mit dieser Funktion können die Geräte, Module, Kanal- und/oder Block-Nummern variabel gehalten werden.
	device ist die Geräte-Nummer, bei nur einem Gerät =0. ***board*** ist die Nummer des TPCX- bzw. TPCE-Moduls, 0=Modul A bis 15=Modul Q.
	input ist die Kanal-Nummer des entspricht Moduls von 1 bis 8.
	blockNr ist die Block-Nummer, 0 bis "NBlks-1" (Funktion), falls Multiblock- oder ECR-Aufnahme, sonst gleich 0.
	Wenn bei ***input*** ein "**@**" vor der Kanal-Nummer steht, wird das entsprechende, durch Mittelung erzeugte, Signal zurück gegeben.
	
	Beispiele:
	`Blk = 0`
	`value = GetChannel(0, 0, 1, Blk)`
	`; Kurve aus Device 0, Gerät 0, c0A1, Block 0`
	
	`avg3 = GetChannel(0, 1, @3, 0)`
	`; Gemittelte Kurve aus Device 0 Gerät 1, @c0B3`

c0A1, c0A2, …. **@c0A1, @c0A2, ...**	Liste aller im System installierten Hardware-Kanäle. Signale, die durch Mittelung erzeugt wurden sind durch ein "@" vor dem Kanal-Bezeichner definiert

26.2 Gruppe "All Functions"

In dieser Gruppe sind alle verfügbaren Funktionen in alphabetischer Reihenfolge aufgelistet.

26.3 Gruppe "Base Functions"

Name [Einheit] = Ausdruck	**Prinzip einer Formel.**

a + b	Addition zweier Werte. ***a*** und ***b*** können eine Zahl, eine Kurve, ein String oder ein Array sein. Zahlen werden addiert und ergeben wieder eine Zahl. Kurven werden addiert und ergeben eine neue Kurve. Strings werden zusammen verkettet.
	
	Die Addition einer Kurve mit einer Zahl ergibt einen entsprechenden Offset der Kurve.
	
	Addition eines Strings und einer Zahl ergibt wieder einen String, wobei die Zahl als String interpretiert wird.
	
	Bei der Addition eines Arrays mit einem Array werden die einzelnen Werte im Array miteinander addiert.
	
	Die Addition eines Arrays mit einer Zahl ergibt einen Offset jedes einzelnen Wertes im Array.
	
	Beispiele:
	Zahlen
	`value = 5+3`
	Das Resultat value ist eine Zahl mit dem Wert 8.
	
	Strings
	`value = "c0A"+1`
	value ist ein String mit dem Wert "c0A1"
	
	`value = "c0A"+(1+1)`
	value ist ein String mit dem Wert "c0A2"
	
	`value = "c0A"+1+1`
	value ist ein String mit dem Wert "c0A11"
	
	Kurven
	`value = c0A1 + c0A2`
	value ist eine neue Kurve
	
	`value = c0A1 + 5`
	value ist die Kurve c0A1 um den Wert 5 verschoben in der Amplituden-Achse. (der Einzelwert entspricht der physikalischen Einheit der Signal-Kurve).
	
	Arrays
	`value = arr1 + 1`
	arr1 sei ein Array vom Typ double mit den werten (0, 2, 5, 7). Dann ist value auch ein Array vom Typ double, mit den Werte (1, 3, 6, 8).

	`value = arr1 + arr2` arr1 und arr2 seien Arrays vom Typ double. arr1 habe die Werte (0, 2, 5, 7), arr2 die Werte (8, 7, -6, -5). Dann ist value ein Array vom Typ double, mit den Werte (8, 9, -1, 2).
a - b	Subtraktion zweier Werte. ***a*** und ***b*** können eine Zahl, eine Kurve oder ein Array sein. Zahlen werden subtrahiert und ergeben wieder eine Zahl. Kurven werden subtrahiert und ergeben eine neue Kurve. **Strings können nicht subtrahiert werden.** Die Subtraktion einer Kurve mit einer Zahl ergibt einen entsprechenden Offset der Kurve. Bei der Subtraktion eines Array mit einem Array werden die einzelnen Werte im Array voneinander subtrahiert. Die Subtraktion einer Zahl von einem Array ergibt eine Subtraktion jedes einzelnen Wertes im Array. **Beispiele:** Analoge Beispiele siehe unter a + b.
a * b	Multiplikation: ***a*** und ***b*** können Zahlen, Kurven, Arrays oder eine Kombination davon sein. Das Resultat ist wieder eine Kurve, wenn mind. ein Argument eine Kurve ist, sonst ist das Resultat wieder ein Einzelwert.
a / b	Division: ***a*** und ***b*** können Zahlen, Kurven, Arrays oder eine Kombination davon sein. Das Resultat ist wieder eine Kurve, wenn mind. ein Argument eine Kurve ist, sonst ist das Resultat wieder ein Einzelwert.
a ^ b	Potenz-Funktion: ***a*** und ***b*** können Zahlen, Kurven oder eine Kombination davon sein. Das Resultat ist wieder eine Kurve, wenn mind. ein Argument eine Kurve ist, sonst ist das Resultat wieder ein Einzelwert. Diese Funktion liefert natürlich nur ein brauchbares Resultat, wenn die Regeln der Potenzierung eingehalten werden. Z.B. ist -3^0.5 nicht definiert für reelle Zahlen.
Sqrt (a)	Berechnet die Quadratwurzel. ***a*** kann entweder eine Zahl oder eine Kurve sein und das Resultat wird wieder eine Zahl oder eine Kurve sein.
Abs (a)	Bildet den Absolutwert (immer positiv). ***a*** kann entweder eine Zahl oder eine Kurve sein und das Resultat wird wieder eine Zahl oder eine Kurve sein. Abs () kann ebenfalls für komplexe Werte verwendet werden, siehe auch die Funktionen Real(), Imag() und Angle()

Diff (a)	Berechnet das Differential der Kurve ***a***. Das Resultat ist wieder eine Kurve.
	Es gilt zu beachten, dass bei der Differentiation der relativ hochfrequente Rauschanteil des AD-Wandlers zu unbrauchbaren Resultaten führen kann. Mit der Anwendung von Filter- oder Smooth-Funktionen kann dieser Effekt unterdrückt werden.
	
	Beispiel:
	`;len ist in Meter skaliert`
	`velocity[m/s] = smooth(Diff(len), 100)`

Int (a)	Berechnet das Integral der Kurve ***a***. Das Resultat ist wieder eine Kurve.
Limit (a, min, max)	Begrenzt ***a*** auf den Bereich ***min*** bis ***max***. Wenn *a* eine Zahl ist, ist das Resultat wieder eine Zahl. Falls *a* eine Kurve ist, wird jeder einzelne Wert begrenzt und das Resultat ist wieder eine Kurve. Diese Funktion wird gebraucht, wenn eine Division durch Null (bzw. nahezu Null, bei Kurven mit Nulldurchgängen) erfolgen kann.
MaxOf (a, b)	***a*** und ***b*** müssen Zahlen sein. Die Funktion liefert als Resultat die grössere von beiden Zahlen.
MinOf (a, b)	***a*** und ***b*** müssen Zahlen sein. Die Funktion liefert als Resultat die kleinere von beiden Zahlen.

Integer (a)	***a*** muss eine Zahl sein. Das Resultat ist ***a*** abgerundet auf die nächste ganze Zahl (bis zu 15 Dezimalstellen).
	Diese Funktion wird vor allem zum Runden von Einzelwerten gebraucht.
	
	Beispiel:
	`; Vorgängigen Berechnung`
	`; mehrere Perioden eines Signals`
	`Phas[°]=682.5`
	
	`; Phasenlage soll +/-180° sein`
	`Phas[°]=360*(Phas/360 - Integer(Phas/360+0.5))`
	`; Phas = -37.5°`

SetRange (a, min, max) **SetRange (a, b)**	***a*** und ***b*** müssen Kurven sein, ***min*** und ***max*** Zahlen. Falls ***b*** angegeben wird, werden ***min*** und ***max*** aus der Kurve ***b*** übernommen (der Amplituden-Bereich von ***a*** wird demjenigen von ***b*** gleichgesetzt). Diese Funktion ist nützlich, wenn Kurven-Diagramme mit reproduzierbaren Y-Skalen erzeugt werden sollen. Sie erleichtert auch den graphischen Vergleich von verschiedenen berechneten oder originalen Signalen untereinander. Bei jeder mathematischen Funktion wird der (virtuelle) Messbereich einer Kurve beeinflusst. Dieser wird durch die minimalen und maximalen Werte innerhalb der Resultat-Kurve festge-

	legt. Der Messbereich kann sich also nach jeder neuen Berechnung ändern.
	Durch ***SetRange*** kann der Resultat-Kurve ein bestimmter und somit reproduzierbarer Messbereich aufgezwungen werden.
	Sinnvollerweise sollte die Funktion ***SetRange*** am Ende der Berechnungen angewendet werden, ansonsten werden dessen Änderungen wieder verworfen.
	
	Beispiel:
	`trace = c0A1 / 10`
	`; Range wird (automatisch) auf berechnete Werte angepasst`
	
	`trace = SetRange(trace, c0A1)`
	`; Range entspricht dem von c0A1`
	`; Weitere Zuweisungen auf trace ändert deren Range wieder!`

GetCrs ("Wname ", "A")	Gibt den Zeitwert in Sekunden der aktuellen Position des Cursors ***A*** auf der Zeitachse des Kurvenanzeigefensters ***Wname*** zurück. ***Wname*** muss eine gültige Fensterbezeichnung, ***A*** eine gültige Cursor-Bezeichnung (A, B, C, ...) sein
SetCrs ("Wname", "A", t)	Setzt den Cursor ***A*** im Kurvenanzeigefenster ***Wname*** auf die Zeit ***t*** (in Sekunden). ***Wname*** muss eine gültige Fensterbezeichnung, ***A*** eine gültige Cursor-Bezeichnung (A, B, C, ...) sein.

26.4 Gruppe "File Functions"

DeleteFile (Filename)	Löscht eine oder mehrere Dateien. Der Parameter ***Filename*** kann dabei ein String, String-Array oder eine Liste mit Dateinamen sein.
	Die Funktion bringt **False** (=0) zurück, wenn mind. eine Datei nicht gelöscht werden konnte (die Datei ist vielleicht im Signalquellen-Browser eingetragen und/oder Kurven davon sind in einer Kurvenanzeige vorhanden).
	True (=gültig) wird auch zurückgegeben, wenn Dateien nicht vorhanden sind.
	
	Beispiel:
	`; Signal generieren und als Datei speichern`
	`tr = Noise(1, 1E3, 1E3)`
	`Save("test.tpc5", tr)`
	`; Kurve wieder löschen`
	`retval = DeleteFile("tests.tpc5")`
	
	`if retval = False then ; Rückgabe prüfen`
	`  SetFormulaError("Konnte nicht gelöscht werden!")`
	`endif`

File (
Filename, Index)
 	
File (
Filename, Index)
[.Block] ['Marker]
	Die Funktion ***File*** gibt eine Kurve zurück. Es können die analogen Signale als auch die digitalen (Marker) Signale ausgelesen werden. Bei Multiblock-Aufnahmen kann auf die einzelnen Blöcke zugegriffen werden. Es wird immer nur ein Block ausgelesen.
	
	Filename ist ein String mit dem Namen der zu öffnenden Datei. Es werden nur Dateien mit der Erweiterung *.tpc5 oder *.tdp akzeptiert. Wenn kein Pfad mit dem Dateinamen angegeben wird, sucht das Programm die Datei im Verzeichnis **DATA** des **aktuellen Experiments**.
	Für eine Datei in einem andern Verzeichnis innerhalb des aktuellen Experiments muss nicht die gesamte Pfadbezeichnung angegeben werden. Es genügt die Bezeichnung ***"..\Ref \NAME.tpc5"***.
	
	Index ist die Kurven-Nummer in der Datei (nicht zu verwechseln mit der Kanal-Nummer). 0 entspricht der ersten Kurve.
	
	Beispiel:
	`File("dateiname.tpc5", 1)`
	In diesem Beispiel wird also die zweite Signalkurve der Datei für die Berechnung verwendet.
	
	Mit ***.Block*** kann bei Multiblock-Aufnahmen auf die einzelnen Blöcke zugegriffen werden. Die Block-Nummer ist fakultativ. Wenn sie fehlt, wird immer Block 0 angenommen. Es wird immer nur ein Block ausgelesen.
	Beispiel:
	`File("Dateiname", Index).Block`
	
	Mit ***'Marker*** kann anstelle der Analog-Kurve (entsprechend dem Parameter ***Index***) eine der beiden digitalen Spuren (Marker) aus-

	gelesen werden. Dieser Parameter kann die Werte 1 oder 2 annehmen. Das Resultat besteht aus einer Kurve mit den Amplitudenwerten 0 oder 1. Marker sind nur vorhanden, wenn bei der ursprünglichen Aufnahme im Control-Panel "Averaging" nicht auf 16Bit gestellt war.
	Beispiel:
	`File("crash.tpc5",0)'1`
	
	Beispiele:
	`signal_a = File("messung.tpc5", 1)`
	Liefert das analoge Signal des zweiten Kanals der Datei "messung.tpc5" zurück.
	
	`signal_m = File("messungblock.tpc5", 0).3'1`
	Liefert das Marker-Signal im vierten Block (beginnend mit 0) der ersten Kurve in der Datei "messungblock.tpc5" zurück.

FileExist (Filename)	***Filename*** ist ein String mit dem Namen der gesuchten Datei. Bezüglich Datei-Pfad gilt das gleiche, wie unter "**File(....)**".
	Der Rückgabewert ist 1 (*True*), wenn die Datei existiert. Ist die Datei nicht vorhanden wird eine 0 (*False*) zurückgegeben.
	
	Beispiel:
	`if FileExist("xy.tpc5") = True then`
	`  Ntrace1=NTraceInFile ("xy.tpc5")`
	`else`
	`  Ntrace=0`
	`endif`

FileIndexExists (Filename, index)	Überprüft ob ein ***index*** (0…) in ***Filename*** existiert. ***Filename*** kann als String der Name einer *.TPC5 Datei oder eine Referenz (z. B. Ref1) sein.
	Siehe auch ***NTracesInFile (Filename)***.
	Für Referenzen kann auch die Funktion ***Length (Filename)*** verwendet werden.
	
	Beispiel 1:
	`fname$ = "test.tpc5"; Signale generieren`
	`tr1 = Noise(1, 1E3, 1E3)`
	`tr2 = Noise(2, 1E3, 1E3)`
	`Save(fname$, tr1, tr2) ; Kurven speichern`
	
	`for a = 0 to 10 step 1 ; hier bis zu 11 Kurven`
	`  val = FileIndexExist (fname$, a)`
	`  if val = True then  ; ist Kurve vorhanden ?`
	`  tr = File(fname$, a) ; Kurve kann geholt werden`
	`  ; weitere Berechnungen`
	`  endif`
	`next`
	
	Beispiel 2:
	`If FileIndexExist(ref1, 2)=True then`
	`  ; weitere Berechnungen`
	`endif`

GetFiles (sourcePath [,nameMask])	Gibt ein Array aus Strings mit den vorhandenen Dateien im Verzeichnis ***sourcePath***. Jedes Element im Array beinhaltet den Na-

	me einer Datei.
	Mit dem optionalen String-Parameter ***nameMask*** kann eine Namen-Maske (z.B. "Test-*.tpc?") vorgegeben werden. Es werden dann nur die Dateinamen aufgelistet, welche mit der Maske übereinstimmen. Ein "?" steht für irgend einen einzelnen Charakter. Mit einem "*" werden auch mehrere unbestimmte Charakter zugelassen.
	
	Beispiel:
	`FilAr$ = GetFiles(".\data", "Test*.tpc5")`
	`Nfil = Length(FilAr$) ; number of files found`

NTracesInFile (
Filename)
	Filename ist ein String mit dem Namen der zu prüfenden Datei. . Es werden nur Dateien mit der Erweiterung *.tpc5, *.tdp oder *.bdf akzeptiert. Bezüglich Datei-Pfad gilt das gleiche, wie unter "File(....)".
	Die Rückgabe ist eine Zahl, welche die Anzahl der enthaltenen Kurven angibt. 0 bedeutet, dass keine Kurven vorhanden sind. Falls die Datei ***Filename*** nicht vorhanden ist, meldet der Formel Editor einen Fehler.
	
	Beispiel:
	`if FileExist("xy.tpc5") = True then`
	`  Ntrace1=NTracesInFile("xy.tpc5")`
	`else`
	`  Ntrace=0`
	`endif`

Save (
Filename, Trace*)
	Filename ist ein String mit dem Namen der zu erzeugenden Datei. Der Dateiname muss die Erweiterung *.tpc5 haben, wenn die gespeicherten Signale wieder mit TranAX analysiert werden sollen.
	Wenn ein "#" am Ende des Dateinamens ist, wird dieses Zeichen durch eine fortlaufende Nummer ersetzt. (z.B. "Check#.tpc5" ergibt "Check001.tpc5", "Check002.tpc5", ...).
	Bezüglich Datei-Pfad gilt das gleiche, wie unter "File(....)".
	Trace bezeichnet die Kurven-Variable die gespeichert werden soll. Es können auch mehrere durch Kommata getrennte Kurven angegeben werden.
	
	Falls die Datei erfolgreich gespeichert werden konnte, gibt die Funktion eine 1 (*True*), im Fehlerfall eine 0 (*False*) zurück.
	Eine bereits existierende Datei mit gleichem Namen wird ohne Warnung überschrieben.
	
	Beispiel:
	`v[m/s] = Smooth(c0A1,200)`
	`Save_OK = Save("Velocity.tpc5", v)`
	`; File may be open`
	`if Save_OK = False then`
	`  ; Use a different name`
	`  Save("Velocity-2.tpc5", v)`
	`endif`

	Mit der Funktion Save() können auch BDF Dateien in TPC5 Dateien konvertiert werden.
	
	BDF (Binäres Daten Format), sind Dateien welche z.B. im Schreiber Modus erstellt werden. Es ist dies ein spezielles Datei-Format. Dieses ermöglicht auch eine grosse Datenmengen mit maximaler Geschwindigkeit auf die Festplatte zu schreiben.
	
	Beispiel:
	Hier werden alle *.BDF Dateien im Ordner "data" des aktuellen Experiment in *TPC5 konvertiert:
	`BDFfiles = GetFiles(".\data", "*.bdf")`
	
	`for i = 0 to Length(BDFfiles)-1 step 1`
	`  fnBDF = BDFfiles (i)`
	`  fnTPC5 = Slice(fnBDF, 0, Length(fnBDF)-4) +`
	`  "tpc5"  ; substitude bdf by tpc5`
	`  save (fnTPC5, fnBDF) ; Save file as *.TPC5`
	`next`

WriteLine (fname ,separator, value*)	Schreibt den/die Werte ***value*** in die Text-Datei ***fname*** (normalerweise im Data-Verzeichnis des aktuellen Experiments). Ein "**#**" hinter dem Datei-Namen wird durch eine fortlaufende Nummer ersetzt. Dadurch kann ein Überschreiben von bestehenden Dateien verhindert werden.
	separator kann ein beliebiger String sein (auch nur ein Einzelzeichen z.B. Tabulator). Damit wird die Trennung zwischen den Werten festgelegt.
	value* sind Einzel-Werte oder Strings. Diese können auch vorher berechnet worden sein (Namen unter "Results"). ***value*** kann auch als Array oder List deklariert werden *).
	Falls die Text-Datei bereits existiert, werden beim ersten Aufruf dieser Funktion alle Einträge ohne Warnung gelöscht. Bei nachfolgenden **WriteLine-**Aufrufen (beim aktuellen Abarbeiten der Formel-Kommandos) wird jeweils eine weitere Linie angehängt. Falls die Text-Datei noch nicht existiert, wird sie unter dem Namen ***fname*** erzeugt.
	*) Bei dieser Funktion kann es unübersichtlich werden, wenn mehr als ein Array bzw. List angegeben ist.
	
	Beispiel:
	`fn$ = "Test-#.txt"`
	
	`for i = 0 to 5 step 1`
	`  ; do some calculations`
	`  var1 = Sin(i)`
	`  var2 = Cos(i)`
	`  var3 = i^2`
	`  ; write the values to the text file`
	`  WriteLine(fn$, Tabulator, var1, var2, var3)`
	`next`

WriteColumn (fname, separator, value*)	Diese Funktion macht im Wesentlichen das gleiche wie **WriteLine()**.

	Die einzelnen Elemente eines Arrays und/oder einer Liste werden dabei in derselben Kolonne abgelegt. Für jeden einzelnen Parameter ***value*** entsteht also eine Kolonne.
	Beim wiederholten Aufrufen von **WriteColumn()** werden die Kolonnen unten mit den neuen Werten erweitert. Da dies meist auch zu Unübersichtlicheit führt, sollte der Aufruf nur einmal erfolgen.
	Auch ist diese Funktion kaum geeignet, wenn nur Einzel-Werte vorkommen. Dies ergäbe den gleiche Datei-Inhalt wie bei WriteLine().
	
	Beispiel:
	`fn$ = "Test-#.txt"`
	`Ar1=Array(0 to 10) as Double`
	`Ar2=Array(0 to 10) as Double`
	`Ar3$=Array(0 to 10) as String`
	`for i = 0 to 10 step 1`
	`  ; do some calculations`
	`  Ar1(i)=i`
	`  Ar2(i)=2*i`
	`  Ar3$(i)=StringFormat(i,"0.00")`
	`next`
	`WriteColumn(fn$, Tabulator, Ar1, Ar2, Ar3$)`

Tabulator **Semicolon** **Comma** **Space**	Schlüsselwörter für den Parameter ***seperator*** der Funktionen **WriteLine()** und **WriteColumn(),** Sie bestimmen das Trennzeichen der einzelnen Einträge.
CloseFile (fname)	Schliesst die Datei mit dem Namen ***fname***. Dies ermöglicht einen Wiederbeginn beim Schreiben in eine Datei (z.B. bei WriteLine(Name_#, ...). Bemerkung: Geöffnete Dateien werden automatisch geschlossen, wenn alle Funktionslinien fehlerfrei abgearbeitet werden konnten.
StoreReadouts (Filename, Results*) **StoreReadouts (Filename, ScalarTableName, Line Or Column, LineNr | ColNr [, Results]*)**	Speichert die durch Formeln ermittelten Einzelwert-Resultate ***Results*** in eine Text-Datei mit dem Namen ***Filename***. Speichert die Resultate aus der Skalar-Tabelle ***ScalarTableName*** in eine Text-Datei mit dem Namen ***Filename***. Mit den Parametern ***Line*** oder ***Column*** und der entsprechenden Nummer ***LinNr*** oder ***ColNr*** können die Zellen in der Skalar-Tabelle bestimmt werden. Optional können auch noch die durch Formeln ermittelten Einzelwert-Resultate ***Results*** angehängt werden. Bezüglich Datei-Pfad gilt das gleiche, wie unter "File(....)" beschrieben. Wenn ein "#" am Ende des Dateinamens ist, wird dieses Zeichen durch eine fortlaufende Nummer ersetzt.
Line	Schlüsselwort, welches für die Funktion **"StoreReadouts"** verwendet wird. Mit ***Line*** wird die entsprechende Linie/Zeile der Skalartabelle gespeichert. **Beispiel:**

	`StoreReadouts("readouts.txt", "Skalar_A 1", Line, 1)`
Column	Schlüsselwort, welches für die Funktion **"StoreReadouts"** verwendet wird. Mit ***Column*** bzw. ***Col*** wird die entsprechende Reihe/Spalte der Skalartabelle abgespeichert. **Beispiel:** `StoreReadouts("readouts.txt", "Skalar_A 1", Col, 1)`

26.5 Gruppe "Signal Analysis"

Die Funktionen in dieser Gruppe bringen in der Regel einen Einzelwert, aus den entsprechenden Eigenschaften einer Kurve retour.

a (t)	***a*** muss eine Kurve sein, ***t*** eine Zeit in Sekunden. Mit Hilfe der Klammern kann ein **Wert aus einer Kurve** an einer bestimmten Stelle ermittelt werden. Das Resultat ist eine Zahl.
	
	Beispiel:
	`value = cA01(0.01)`
	`; value: Y-Wert zum Zeitpunkt 10ms`

Max (a)	***a*** muss eine Kurve sein. Die Funktion sucht den **grössten Amplitudenwert** von ***a***. Das Resultat ist eine Zahl.
Min (a)	***a*** muss eine Kurve sein. Die Funktion sucht den **kleinsten Amplitudenwert** von ***a***. Das Resultat ist eine Zahl.
TOfMax (a)	***a*** muss eine Kurve sein. Die Funktion liefert den x-Wert (**Zeit in Sekunden**), des **grössten Amplitudenwerts** von ***a***, relativ zum Triggerzeitpunkt wenn die Aufnahme im Scope-Mode erfolgt ist oder relativ zum Zeitpunkt des Startkommandos bei Aufnahmen im Multiblock-, Kontinuierlichen- oder ECR-Mode. Das Resultat ist eine Zahl.
TOfMin (a)	***a*** muss eine Kurve sein. Die Funktion liefert den x-Wert (**Zeit in Sekunden**), des **kleinsten Amplitudenwerts** von ***a***, relativ zum Triggerzeitpunkt wenn die Aufnahme im Scope-Mode erfolgt ist oder relativ zum Zeitpunkt des Startkommandos bei Aufnahmen im Multiblock-, Kontinuierlichen- oder ECR-Mode. Das Resultat ist eine Zahl.
Mean (a)	***a*** muss eine Kurve sein. Berechnet den **Mittelwert** über die ganze Kurve. Das Resultat ist eine Zahl.
MeanP (a)	***a*** muss eine Kurve sein. Berechnet den **Mittelwert** über eine oder mehrere ganze **Perioden**. Das Resultat ist eine Zahl.
RMS (a)	***a*** muss eine Kurve sein. Berechnet den **Effektivwert** über die ganze Kurve. Das Resultat ist eine Zahl.
RMSP (a)	***a*** muss eine Kurve sein. Berechnet den **Effektivwert** über eine oder mehrere ganze **Perioden**. Das Resultat ist eine Zahl.
Freq (a) **Freq (** **a, level, hysteresis)**	***a*** muss eine periodische Kurve sein. Berechnet aus den Nulldurchgängen der Kurve die **mittlere Frequenz**. Das Resultat ist eine Zahl (in Hz). Der Pegel und die Hysteresis für die Nulldurchgänge werden automatisch ermittelt (Pegel = Mitte zwischen Minimal- und Maximal-Wert, Hysteresis = 20% (+/-10%) von Peak-Peak. ***level*** (Pegel) und ***hysteresis*** (in der Einheit der Kurve) können auch explizit angegeben werden. In diesem Fall wird der Automatismus unterdrückt..
TCross (a, tStart, **tStop, level)**	***a*** muss eine Kurve sein, ***tStart*** und ***tStop*** müssen Zeiten in Sekunden sein. ***level*** ist ein Amplitudenwert. Die Funktion sucht begin-

	nend von ***tStart*** in Richtung ***tStop*** (geht vorwärts und rückwärts) die nächste Stelle, bei der ***a*** den Wert ***level*** über- oder unterschreitet. Das Resultat ist ein Zeitwert in Sekunden. Falls im definierten Suchbereich **kein Kreuzen** der Kurve ***a*** mit dem Pegel ***level*** stattfindet, wird als Resultat "n.def" ausgegeben.
	
	Beispiel:
	`t1 = TCross(c0A1,-1,1,4.2)`
	
	`if t1 = NotDefined then`
	`  ; An den linken Rand setzen`
	`  t1 = -1`
	`endif`
	
	`; Platziere Cursor A auf t1`
	`SetCrs("Waveform 1","A",t1)`

TBegin (a)	***a*** muss eine Kurve sein. Die Funktion liefert den x-Wert (**Zeit in Sekunden des ersten Samples** von ***a***, relativ zum Triggerzeitpunkt wenn die Aufnahme im Scope-Mode erfolgt ist oder relativ zum Zeitpunkt des Startkommandos bei Aufnahmen im Multiblock-, Kontinuierlichen- oder ECR-Mode. Das Resultat ist eine Zahl.
TEnd (a)	***a*** muss eine Kurve sein. Die Funktion liefert den x-Wert (**Zeit in Sekunden des letzten Samples** von ***a***, relativ zum Triggerzeitpunkt wenn die Aufnahme im Scope-Mode erfolgt oder relativ zum Zeitpunkt des Startkommandos bei Aufnahmen im Multiblock-, Kontinuierlichen- oder ECR-Mode). Das Resultat ist eine Zahl.
TSample (a)	***a*** muss eine Kurve sein. Die Funktion liefert die **Abtastperiode** (Zeit in Sekunden).
TTrigger (a)	***a*** muss eine Kurve sein. Die Funktion liefert den x-Wert (**Trigger-Zeitpunkt**) von ***a*** in Sekunden, relativ zum Zeitpunkt des Startkommandos bei Aufnahmen im Multiblock- oder ECR-Mode. Bei Aufnahmen im Scope- oder Kontinuierlichen-Mode ist das Resultat = 0. Das Resultat ist eine Zahl.
GetNrOfSamples(a, t0, t1)	***a*** muss eine Kurve sein, ***t0*** und ***t1*** sind Zeiten in Sekunden. Die Funktion liefert die Anzahl Samples im entsprechenden Zeitbereich der Kurve. Falls **t1 < t0**, wird ein negativer Wert geliefert.

GetDate (a)	***GetDate*** liefert Datum und Zeit der Aufnahme der Kurve .
	
	Der Rückgabewert ist eine Zahl im Format double, welche Datum- und Zeit-Angaben enthält. Um die einzelnen Werte zu erhalten, können die folgenden Funktionen verwendet werden (siehe auch "TranAXcommonFunctions.fnc"):
	
	Beispiel:
	`date = GetDate(c0A1)`
	
	`date = date / 10E3`
	`year = Integer(date)`

	`date = date - year`
	`date = date * 100`
	`month = Integer(date)`
	`date = date - month`
	`date = date * 100`
	`day = Integer(date)`
	`date = date - day`
	`date = date * 100`
	`hour = Integer(date)`
	`date = date - hour`
	`date = date * 100`
	`minute = Integer(date)`
	`date = date - minute`
	`date = date * 100`
	`sec = Integer(date)`

SetDate (a [, b])	***a*** und ***b*** müssen Kurven sein, ***b*** ist optional. Das Resultat ist eine Kopie von ***a*** mit geändertem Aufnahmedatum/Zeit. Das Aufnahmedatum wird von ***b*** genommen, wenn ***b*** angegeben ist. Damit kann das Datum und die Uhrzeit einer berechneten Kurve auf den Aufnahmezeitpunkt einer bestimmten Kurve gelegt werden. Wenn ***b*** fehlt, wird die aktuelle PC-Uhrzeit (Zeitpunkt der Berechnung) verwendet.
GetChName(a)	Liefert den Namen der Kurve ***a*** als String. **Beispiel:** `Name1$= GetChName(c0A1)`

GetUnit (a)	Liefert den Namen der physikalischen Einheit der Kurve ***a*** als String.
	Siehe auch **SetUnit()**
	
	Beispiel:
	`; calculations normally will not set a unit`
	`Tr = Smooth(c0A1, 100)`
	
	`Unit$ = GetUnit(c0A1) ; get original unit`
	`SetUnit(Tr, Unit$) ; it will be set in Tr`

SetUnit(a, unit)	Setzt die physikalische Einheit der Kurve ***a*** auf den String ***unit***.
	Siehe auch **GetUnit()**
	
	Beispiel:
	`; calculations normally will not set a unit`
	`Tr = Smooth(c0A1, 100)`
	
	`Unit$ = GetUnit(c0A1) ; get original unit`
	`SetUnit(Tr, Unit$) ; it will be set to Tr`

26.6 Gruppe "Signal Processing"

Die Funktionen in dieser Gruppe geben in der Regel eine entsprechend umgerechnete Kurve zurück.

Shift (a, time) **Shift (a, trace)**	***a*** ist eine Kurve, ***time*** eine Zeit in Sekunden. Das Resultat ist wieder eine Kurve, um ***time*** Sekunden gegenüber dem Trigger-Zeitpunkt **geschoben**. D.h. der Startzeitpunkt (TBegin) wird neu gesetzt. Die Verschiebung wird nicht auf ganze Samplings gerundet. Von Beginn her betrachtet liegen die Stützwerte der neuen und alten Kurve also nicht mehr zwingend übereinander. Dies gilt es zu beachten, wenn bei weiteren Auswerte-Operationen der Beginn und nicht der Triggerzeitpunkt als Referenz genommen wird. Falls anstelle ***time*** eine Kurve (***trace***) angegeben wird, so schiebt sich die Resultat-Kurve in den Zeitbereich von ***trace***. Die Trigger der beiden Kurven stehen dann übereinander.
Slice (a, t0, t1) **Slice (a, t0, t1 [,y0 [,y1]])** **Slice(Array or List, StartIx, Length)**	***a*** muss eine Kurve sein, ***t0*** und ***t1*** zwei Zeiten in Sekunden, ***y0*** und ***y1*** Amplitudenwerte (in V oder Einheit). Die Parameter ***y0*** und ***y1*** sind **optional.** Die Funktion **schneidet den Bereich** zwischen ***t0*** bis ***t1*** von ***a*** aus. Ist ***t0*** **< TBegin(*a*)** bzw. ***t1*** **> TEnd(*a*)** wird die Resultat-Kurve an der entsprechenden Stelle verlängert, falls mindestens ***y0*** angegeben wird. Falls nur ***y0*** angegeben ist, wird die Kurve am Anfang und/oder Ende mit dem konstanten Amplitudenwert ***y0*** verlängert. Falls auch ***y1*** angegeben wird, verlängert sich die Kurve mit dem Amplitudenwert ***y1*** am Anfang (falls ***t0*** **< TBegin(*a*)**) und ***y0*** am Ende (falls ***t1*** **> TEnd(*a*)**). Das Resultat ist eine verkürzte oder verlängerte Kurve. Der Trigger-Zeitpunkt entspricht demjenigen der Kurve ***a.*** Die Slice Funktion kann auch auf ein ***Array*** oder eine Liste ***List*** angewendet werden. ***StartIx*** (>=0) ist das erste der Elemente, ***Length*** ist die Anzahl der Elemente die ausgeschnitten werden.
Skip (a, n)	***a*** muss eine Kurve sein, ***n*** eine Zahl. Diese Funktion bewirkt ein **Resampling**, sodass die Resultat-Kurve entsprechend weniger oder mehr Abtastwerte aufweist. Falls *n* eine positive Zahl ist entsteht eine Resultat-Kurve mit ***n*** mal weniger Abtastwerten. Falls ***n*** negativ ist, werden Abs(***n***) mal mehr Samplings erzeugt. n kann auch eine gebrochene Zahl sein (auch > 1). Zwischenwerte werden durch lineare Interpolation ermittelt. Diese Funktion, ev. kombiniert mit Filtern, erlaubt eine Datenreduktion, die dann sinnvoll ist, wenn Signale mit Überabtastung aufgezeichnet wurden und eine rechenintensive Nachauswertung (z.B. die Funktion "Correlation" oder FFT) gewünscht wird. Der Trigger-Zeitpunkt entspricht demjenigen der Kurve ***a.*** Siehe auch die Funktionen *Resampling()* und *DataReduction()*
Resampling (a,	ändert die Abtastrate der Kurve ***a*** auf die Frequenz ***samplerate***

 © Elsys AG

samplerate)	(in Hz). Ist ***samplerate*** grösser als die ursprüngliche Abtastrate von ***a***, werden die fehlenden Punkte linear interpoliert.
	Ist sie kleiner, werden Punkte entfernt.
	Möglicherweise wird dadurch der Triggerzeitpunkt leicht verschoben, da dieser nach der Operation auf das nächst gelegene Sampling gesetzt wird.
	Siehe auch die Funktion Skip().
	
	Beispiel:
	`tr_orig = c0A1`
	`OrigSR = 1/TSample(tr_orig)`
	`NewSR= 3.1234 * OrigSR`
	`tr_new = Resampling(tr_orig, NewSR)`

SetXAxis (Trace, RefTrace)	Erzeugt eine Kurve mit den Amplituden-Werten von ***Trace*** und den Zeitachsen-Werten der Kurve ***RefTrace***. Werte am Anfang und am Ende der Resultatkurve werden, je nach Überlappung abgeschnitten oder mit Null aufgefüllt. Falls die Abtastraten unterschiedlich sind, erfolgt ein Resampling.
	Trace ist eine Kurve, deren Amplituden-Werte übernommen werden.
	RefTrace ist eine Kurve, deren Zeitachsen-Werte (Abtastrate, Trigger-Delay, Beginn, Ende usw.) übernommen werden.
	
	Beispiel:
	`y = File("messung.tpc5", 0)`
	`x = c0A3.0`
	`z = SetXAxis(y, x)`

SetTrigger (a, t)	***a*** ist eine Kurve, ***t*** eine Zeit in Sekunden. Die Funktion setzt den Trigger-Zeitpunkt der Kurve ***a*** auf den Wert ***t***. Das Verhalten ist wie wenn die **Triggerverzögerung manipuliert** worden wäre. D.h. TBegin und TEnd bleiben auf den ursprünglichen Werten stehen. **Beispiel:** `tr = SetTrigger(tr, 0.001)`
SetTSample (a, t)	***a*** ist eine Kurve, ***t*** eine Abtastrate in Sekunden. Die Funktion liefert die gleiche Kurve ***a***, aber mit der Abtastperiode ***t***. Die Zeitachse der Signal-Kurve wird verändert.
FrqDemod (a) **FrqDemod (a [,Base, [Hyst]])**	***a*** ist eine Kurve. Das Resultat ist wieder eine Kurve. Die Funktion sucht Nulldurchgänge von ***a*** und berechnet daraus an den einzelnen Stellen die Frequenz. Mit den **fakultativen** Parametern ***Base*** und ***Hyst*** kann die Basislinie und die Hysteresis eingestellt werden. Diese Werte bestimmen dann den Suchalgorithmus für die Nulldurchgänge.
Phase (a, b)	***a*** und ***b*** müssen Kurven sein. Das Resultat ist wieder eine Kurve (Einheit: Grad). Die Funktion sucht Nulldurchgänge von ***a*** und ***b*** und berechnet daraus an den einzelnen Stellen die **Phasenverschiebung**.
StdDev (a, width)	***a*** muss eine Kurve sein, ***width*** eine Zahl. Die Funktion berechnet jeweils aus ***width*** Abtastwerten die **Standardabweichung**. Das Resultat ist wieder eine Kurve.
Correlation (	***Trace1*** und ***Trace2*** müssen Kurven sein. Das Resultat ist wieder

Trace1, Trace2)	eine Kurve aus der errechneten Korrelation. Falls ***Trace1*** und ***Trace2*** die gleichen Kurven sind entspricht dies einer *AutoKorrelations-Funktion*.

RegressionPoly (a, n [, bool])	***a*** ist eine Kurve, ***n*** eine ganze Zahl (0 .. 9). Das Resultat ist eine Kurve die der **Polynom-Funktion n-ten grades** entspricht (z.B. Polynome mit n=2 ==> (k0+k1*a+k2*a^2) k0, k1 und k2 werden durch entsprechende Regression ermittelt). Die Resultat-Kurve weist keine grossen Sprünge und Unstetigkeiten mehr auf.
	Wenn der optionale dritte Parameter ***bool*** auf ***True*** (=1) gesetzt wird, liefert die Funktion ein **Array** mit den Polynom-Koeffizienten k0, k1, k2, ... anstelle einer Kurve zurück.
	
	Beispiel:
	`pressure[bar] = File("fielname.tpc5", 0)`
	`; Kurve`
	`poly[bar] = RegressionPoly(pressure, 3)`
	`; Koeffizienten`
	`CoefArr= RegressionPoly(pressure, 3, True)`

ExponentialFit (a [, bool])	***a*** muss eine Kurve sein.
	Die Funktion erzeugt eine Exponential-Kurve, die optimal zu ***a*** passt.
	Bei fallender Kurve im positiven Bereich oder steigender Kurve im negativen Bereich, nimmt die Funktion an, dass ***a*****-->0 bei x--> +∞**.
	Wenn die Verhältnisse umgekehrt sind wird angenommen, dass ***a*****-->0 bei x--> -∞**.
	
	Der Parameter ***bool*** ist optional. Wird dieser auf **True** gesetzt, erhält man anstelle einer Kurve ein Array mit den Koeffizienten y0 und x0, gem. $f(x) = y0 * e^{x0*x}$.
	CoefArr(0)=y0 (=Amplituden-Wert bei x=0)
	CoefArr(1)=x0
	Aus x0 kann die Zeitkonstante ermittelt werden (τ = -1/x0).
	
	Beispiel:
	`; trc = any Trace`
	`xFit=ExponentialFit(trc)`
	
	`xFitAry= ExponentialFit(trc, True)`
	`Tau=-1 / xFitAry(1)`

Spline(points,order, samplerate [, tExtrapolAhead [,tExtrapolBehind]])	Legt einen Polynomzug an die gegebenen Wertepaare ***points***. Ein Wertepaar besteht aus einem Zeitwert auf der X-Achse und dem dazugehörigen Amplitudenwert. Die Wertepaare (**points**) können auch in einem Array deklariert sein. Der Parameter ***order*** ist zur Zeit fix auf 3 (cubic spline) gesetzt, andere Werte werden ignoriert. ***samplerate*** ist die Abtast-Frequenz der Resultat-Kurve. ***tExtrapolAhead*** und ***tExtrapolBehind*** sind zwei fakultative Zeit-Parameter (positive Werte, in Sekunden). Die berechnete Kurve wird vorn und hinten um die angegebenen Zeit-Werte erweitert.

	Beispiel:
	`; create a triangle by x,y value pairs`
	`arr=Array(0,0, 0.05,0, 0.1,1, 0.2,0, 0.25,0)`
	`tr = CreateSignal(1, 1e3, arr)`
	`tr sp = Spline(arr, 3, 1e3, 0.02, 0.015)`

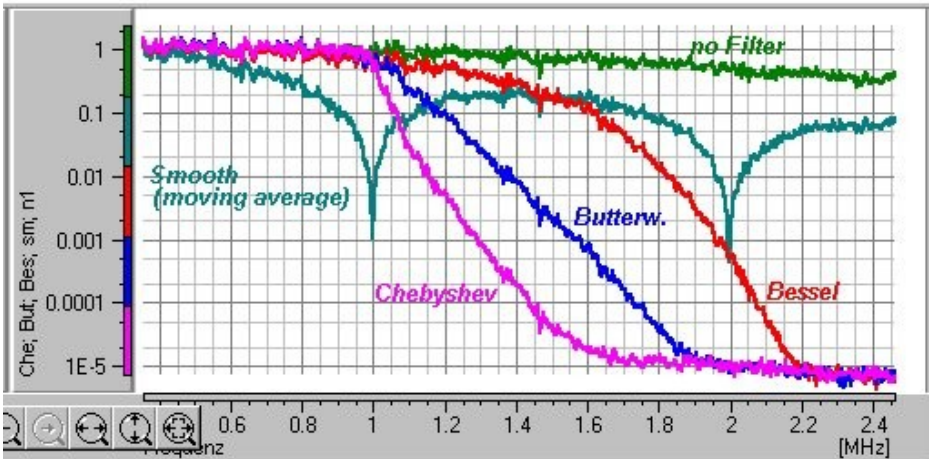
DataReduction(a, type, factor)	Die Datenmenge der Kurve ***a*** wird verringert. ***type*** beschreibt das Verfahren (Skipping , Averaging und MinMaxEnvelope), ***factor*** bestimmt die Reduktion (10 bedeutet eine Reduktion um den Faktor 10). **MinMaxEnvelope** erzeugt eine Zig-Zag-Kurve. D.h. es werden Wertepaare erzeugt, zuerst der Minimal- dann der Maximal-Wert innerhalb des Reduktions-Bereichs. Die Sampling-Zeit wird entsprechend angepasst.
	
	Beispiel:
	`tr1 = c0A1 ; Kurve erfassen`
	`TrRed=DataReduction(tr1, Averaging,10)`

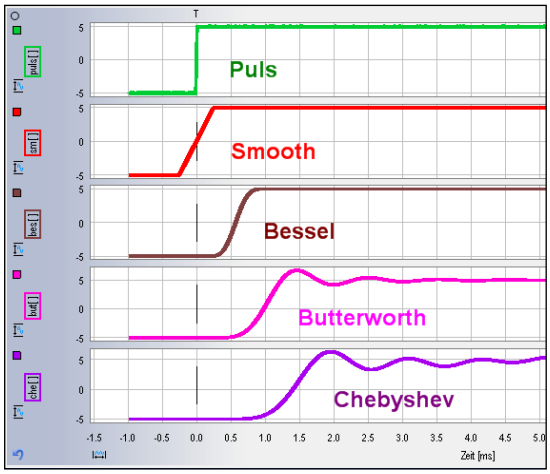
Skipping	Schlüsselwort für ***Type*** bei der Funktion ***DataReduction()***
Averaging	Schlüsselwort für ***Type*** bei der Funktion ***DataReduction()***
MinMaxEnvelope	Schlüsselwort für ***Type*** bei der Funktion ***DataReduction()***

26.7 Gruppe "Filter Functions"

Smooth (a, width)	***a*** muss eine Kurve sein, ***width*** eine Zahl. Die Funktion berechnet den **gleitenden Mittelwert** von ***a*** über ***width*** Abtastwerte. Das Resultat ist wieder eine Kurve.
Median (a, width)	***a*** muss eine Kurve sein, ***width*** eine Zahl. Die Funktion berechnet jeweils aus ***width*** Abtastwerten den Median. Das Resultat ist wieder eine Kurve. Mit dieser Funktion können einzelne (Stör-)Spitzen aus einer Kurve eliminiert werden.
LowPass (a, type [, ripple], order, freq)	***a* ist die Kurve**, die gefiltert wird. ***type*** spezifiziert den **Filtertyp**: *Bessel*, *Butter* (Butterworth) oder *Cheby* (Tschebyscheff). Bei **Tschebyscheff** muss noch der Parameter ***ripple*** (Welligkeit im Durchlassbereich in dB) angegeben werden, ***order*** spezifiziert die **Ordnung** (1 .. 9) des Filters. ***freq*** ist die **Grenzfrequenz** in Hz für Hoch- und Tiefpassfilter resp. die Mittenfrequenz für Bandpass und Bandsperre. Bei letzteren muss auch noch die **Bandbreite** im Parameter ***bandwidth*** angegeben werden.
HighPass (a, type [, ripple], order, freq)	
BandPass (a, type [, ripple], order, freq, bandwidth)	
BandStop (a, type [, ripple], order, freq, bandwidth)	
Bessel	Schlüsselwort zu ***type*** bei den Filter-Funktionen
Butter	Schlüsselwort zu ***type*** bei den Filter-Funktionen
Cheby	Schlüsselwort zu ***type*** bei den Filter-Funktionen

Die folgenden Kurven zeigen einige Beispiele der verschiedenen Filter-Typen:





Chebyshev- und Butterworth-Filter haben zwar eine **grosse Steilheit** im Dämpfungsbereich, sie sind aber **nicht geeignet**, wenn eine möglichst **unverzerrte** Pulsantwort gewünscht wird.

26.8 Gruppe "Programming Functions"

if i = b then
BLOCK1
[else
BLOCK2]
endif
	i und ***b*** sind Variablennamen vom Typ Zahl oder String.
	Anstelle von "**=**" sind auch folgende Vergleichs-Operationen möglich: "**>**", "**<**", "**>=**", "**<=**", "**<>**"
	
	Je nach Ergebnis des Vergleichs werden die Funktionen in ***BLOCK1*** oder ***BLOCK2*** durchgeführt. ***else*** und ***BLOCK2*** sind optional.
	
	Beispiel:
	`if Mx > Bigest then`
	`  Bigest=Mx`
	`else`
	`  Bigest = 0`
	`endif`

for i = iStart to iEnd
[step Increment]
 	
BLOCK
 	
next
	i ist ein Variablen-Name vom Typ Zahl. ***iStart***, ***iEnd*** sind Zahlen, welche auch negativ sein können. ***Increment*** ist ebenfalls eine Zahl, welche aber positiv sein muss.
	
	Die Funktionen in ***BLOCK*** werden in einer Schlaufe so oft durchlaufen, bis ***i > iEnd*** ist. Bei jedem Durchlauf wird ***i*** um den Wert ***Increment*** verändert.
	
	Beispiel:
	`Bigest = -1000`
	`i = 0`
	`for i = 0 to NBlks(c0A1)-1`
	`  Mx = Max(c0A1.i)`
	`  if Mx > Bigest then`
	`  Bigest=Mx`
	`  endif`
	`next`

exitloop	Mit dieser Operation kann eine **Loop-Schlaufe** verlassen werden.
	
	Beispiel :
	`for i=1 to 4 step 1`
	`  if i=3 then`
	`  exitloop`
	`  endif`
	`next`

do until i= b
BLOCK
loop
	i und ***b*** sind Variablen-Namen vom Typ Zahl.
	Anstelle von "**=**" sind auch folgende Vergleichs-Operationen möglich: "**>**", "**<**", "**>=**", "**<=**", "**<>**"
	Die Schlaufe ***do until loop*** wird solange wiederholt, bis die Bedingung ***i=b*** erfüllt (*True*) ist. Falls die Bedingung bereits zu Beginn erfüllt ist, wird der Code (***BLOCK***) zwischen ***do until*** und ***loop*** nicht ausgeführt.
	
	Beispiel:
	`i = 100`
	`do until i = 15`
	`  i = i-1`
	`loop`

do	***i*** und ***b*** sind Variablen-Namen vom Typ Zahl.

BLOCK
loop while i = b
	Anstelle von "=" sind auch folgende Vergleichs-Operationen möglich: ">", "<", ">=", "<=", "<>"
	Die Schlaufe ***do loop while*** wird solange wiederholt, bis die Bedingung ***i=b*** nicht mehr erfüllt (*False*) ist. Der Code (***BLOCK***) zwischen ***do*** und ***loop while*** wird Mindestends einmal ausgeführt.
	
	Beispiel:
	`i = 0`
	`do`
	`  i = i + 1`
	`loop while i < 100`

do while i=b
BLOCK
loop
	i und *b* sind Variablen-Namen vom Typ Zahl.
	Anstelle von "=" sind auch folgende Vergleichs-Operationen möglich: ">", "<", ">=", "<=", "<>"
	Die Schlaufe ***do while loop*** wird solange wiederholt, bis die Bedingung ***i=b*** nicht mehr erfüllt (*False*) ist. Falls die Bedingung bereits zu Beginn erfüllt ist, wird der Code (***BLOCK***) zwischen ***do while*** und ***loop*** nicht ausgeführt.
	
	Beispiel:
	`i = 100`
	`do while i > 15`
	`  i = i - 1`
	`loop`

GetVariableValue (VariableName)	Liefert den Wert der Variablen mit dem Namen ***VariableName*****.**
	VariableName muss ein String sein.
	Falls die Variable nicht vorhanden ist, wird ***NotDefined*** zurück gegeben.
	Diese Funktion wird meistens nach der Funktion **SetVariableValue** benötigt.
	
	Beispiel:
	`SetVariableValue("val_a", 20)`
	`tmp = GetVariableValue("val_b")`
	
	`if tmp = NotDefined then  ; "val_b" vorhanden ??`
	`  tmp = GetVariableValue("val_a")`
	`endif`

SetVariableValue (VariableName, a)	Setzt oder erstellt eine Variable mit dem Namen ***VariableName***.
	Falls bereits eine gleichnamige Variable existiert wird diese auf den neuen Wert gesetzt.
	VariableName muss ein String sein.
	Der Parameter ***a*** kann vom Typ String, Double, Array, List oder Trace sein. Die gesetzte oder erstellte Variable ist vom gleichen Typ wie ***a***
	Um den Wert einer gebildeten Variablen zu erhalten, muss die Funktion **GetVariableValue(*VariableName*)** verwendet werden.
	
	Beispiel 1:
	`; Erstellt die Variablen "val_0" bis "val_5"`
	`; mit dem Initialwert 0`
	`for i=0 to 5 step 1`
	`  SetVariableValue("val_"+i, 0)`

	next
	
	Beispiel 2:
	; Angenommen: Multiblock-Aufnahme auf Datei "xy.tpc5"
	for blk=0 to 5 step 1 ; Blöcke 0 .. 5
	  TrcBlk=File("xy.tcp5", 0).Blk ;Kurve aus Block blk
	  SetVariableValue("MaxBlk_"+blk, Max(TrcBlk))
	next
	MaxOfBlk2=GetVariableValue("MaxBlk_2")

EndFormula	Diese Operation dient als Abschluss der Berechnungen. Wenn sie fehlt werden alle Formel-Linien bis zuunterst abgearbeitet. Sie ist vor allem nützlich bei der Entwicklung von Formel-Programmen. Man kann damit verhindern, dass nachfolgende (ev. noch fehlerhafte) Formeln gelöscht oder auskommentiert werden müssen.
Using "filename"	Das Schlüsselwort ***using*** wird dazu verwendet, eine Funktionsdatei (in der Regel *.fnc) ***filename*** einzubinden. **Beispiel:** Es sei die Datei "MathOperators.fnc" mit dem folgenden Inhalt erstellt worden: Function Addition(value0, value1) Addition = value0 + value1 EndFunction Diese Datei kann nun mit using eingebunden werden: using "MathOperators.fnc" ;Anwendung der Funktion "Addition" result=Addition(2,3)
Function name (parameter*) **BLOCK** **EndFunction**	***Function*** ist ein Schlüsselwort, welches den Anfang eines Formel-Blocks definiert. Nach dem Schlüsselwort wird der Name ***name*** der Funktion sowie die zu übergebenden ***parameter*** in Klammern erwartet. Die Anzahl der Parameter ist nicht limitiert. ***EndFunction*** schliesst die Funktion wieder ab. Es sind alle Variablen-Typen (Zahl, Kurve, Array, Liste usw.) als Parameter zulässig. **Achtung:** Die Parameter werden als *Referenz* übergeben. Sie können also in einer Funktion direkt neu berechnet und zurück gegeben werden. Wenn dies geschieht, **gilt zu beachten**, dass diese Variablen auch im **Haupt-Programm verändert** werden. Der direkte Rückgabewert muss dem Namen ***name*** der Funktion zugewiesen werden. Als Rückgabewert sind alle Variablen-Typen (Zahl, Kurve, Array, Liste usw.) zulässig. Siehe auch die Funktionen ***Using()*** und ***Endformula***. **Beispiel:** ; Function in der selben Formel x = 5 y = 7 z = Addition(x,y) Endformula

	``` Function Addition(value0, value1) Addition = value0 + value1 EndFunction ```
SplitString (text, symbol)	Trennt den String ***text*** mit dem Zeichen ***symbol*** und liefert ein Array mit den jeweiligen Teilstrings (das Zeichen ***symbol*** existiert nicht mehr). ***Symbol*** kann auch aus mehreren Zeichen bestehen. Der ursprüngliche String ***text*** wird dann an der entsprechenden Stelle getrennt. Falls ***symbol*** mehrfach vorkommt in ***text,*** wird dieser in mehrere Teil-Strings gewandelt. Das Array ist dann entsprechend länger. **Beispiel:** ``` text = "trace_001" symbol = "_" StrArray = SplitString(text, symbol) stringTrace = StrArray (0) ; "trace" stringNumber = StrArray (1) ; "001" ```
StringToNumber (text)	Wandelt den String ***text*** in eine Zahl. Vorausgesetzt ist natürlich dass es sich um eine Zahl im String handelt. Ansonsten wird ***Not-Defined*** zurück geliefert. **Beispiel:** ``` Str$ = "-1234" val = StringToNumber(Str$) val = val * 2 ```

True	Schlüsselwort ***True***, entspricht dem Wert 1. Es signalisiert, dass eine Bedingung erfüllt ist.
	
	Beispiel:
	``
	if FileExist("xy.tpc5") = True then
	  BLOCK
	endif
	``

False	Schlüsselwort ***False***, entspricht dem Wert 0. Es signalisiert, dass eine Bedingung nicht erfüllt ist.
	
	Beispiel:
	``
	if FileExist("xy.tpc5") <> False then
	  BLOCK
	endif
	``

NotDefined	Schlüsselwort, Rückgabewert von Funktionen, welche keinen gültigen Wert ermitteln konnten.
	
	Beispiel:
	``
	tx=TCross(trc,-10,10,0.5)
	If tx = NotDefined Then
	 ; Kein Übergang gefunden
	Endif
	``

SetEnvironment (key, bool)	Definiert das Verhalten bei der Berechnung von gewissen Funktionen. Damit lassen sich Prozesse bezüglich Geschwindigkeit / Speicherplatz optimieren. ***bool*** (True oder False) schaltet die Optimierungsart ein oder aus. ***key*** kennzeichnet die Optimierungsart:

	***key* = 0**: Kurven-Resultate (Traces) werden im Arbeitsspeicher und nicht als Datei im Verzeichnis " \expr\" abgelegt. Dadurch können die Berechnungszeiten wesentlich verkürzt werden. Andererseits wird dazu mehr Arbeitsspeicher belegt.
	
	Beispiel:
	`SetEnvironment(0,True) ; Einschalten`
	`Trc2 = c0A2 ; Trc2 ist im Arbeitsspeicher`
	
	`SetEnvironment(0,False) ; Ausschalten`
	`Trc3=c0A3 ; Trc3 ist als Datei vorhanden`

StringFormat (value, "format")	***Stringformat*** konvertiert eine Zahl ***value*** in einen String. Dessen Formatierung ***format*** kann als String definiert werden.
	
	Beispiele:
	`N$ = StringFormat(1000, "0.00 V")`
	`; N$ = "1000.00 V"`
	
	`Findex=3`
	`Fix$= StringFormat(Findex, "000")`
	`; Fix$="003"`

WriteForNext (key, value)	***WriteForNext*** dient zusammen mit ***ReadOfPrevious*** als eine Art globale Variable. ***key*** ist der Name der Variable als String. ***value*** kann eine Zahl, String, Array, List oder Trace sein. Für mehr Informationen und Beispiel siehe die Funktion ***ReadOfPrevious()***.

ReadOfPrevious (key)	***ReadOfPrevious*** dient zusammen mit WriteForNext als seine Art globale Variable. ***key*** ist ein String und der Name der Variable.
	
	Auf die in der Variablen gespeicherten Werte kann bei jedem erneuten Durchlauf der Formeln zugegriffen werden. So können Berechnungen über mehrere Messungen hinweg durchgeführt werden.
	
	Beispiel:
	`; Mittelwertbildung`
	`; Letzte Kurve laden`
	`mytrace = ReadOfPrevious("mytrace")`
	
	`; Neue Messung erfassen und glätten`
	`tr = c0A1;`
	`tr = Smooth(tr, 100)`
	
	`; Prüfen, ob erster Durchlauf`
	`if mytrace = NotDefined then`
	`  ; Ja, erster durchlauf`
	`  mytrace = tr`
	`else`
	`  ; Nein, nicht der erste Durchlauf`
	`  mytrace = (mytrace + tr) /2`
	`endif`
	
	`; Wert wieder zwischenspeichern`
	`WriteForNext("mytrace",mytrace)`

Delay (seconds)	Verzögert den Ablauf entsprechend dem Parameter ***seconds***. **Bemerkung:** Diese Funktion liefert keine genaue zeitliche Verzögerung und sollte deshalb nicht für genaue Zeitmessungen eingesetzt werden. Die Abweichungen hängen von parallel laufenden Prozessen im Betriebssystem ab. Diese können einige ms betragen.

26.9 Gruppe "Array Functions"

array (min to max) as double
 	
array (min to max) as string
 	
array (min to max) as complex
	Erstellt ein Array, welches Werte vom Typ ***double***, ***string*** oder ***complex*** enthalten kann. Ein Array kann nur einen Datentyp beinhalten. ***min*** und ***max*** sollten ganzzahlig sein. Sie definieren die Grösse des Arrays. "0 to 3" ergibt vier Felder, "7 to 12" ergibt sechs Felder.
	
	Auf die einzelnen Werte kann mit Klammern ***array (n)*** zugegriffen werden. ***n*** muss dabei innerhalb der Werte liegen, in der das Array definiert wurde.
	
	Beispiel:
	`arr1 = Array(0 to 3) as double`
	`arr2 = Array(0 to 3) as String`
	
	`; Erstes Feld auslesen`
	`number = arr1(0)`
	
	`; Zweites Feld auf "D" setzen`
	`arr2(1) = "D"`

Double	Schlüsselwort für Zahlenformat einer Variablen (doppelte Genauigkeit, 64Bit), normalerweise zur Deklaration von Arrays.
String	Schlüsselwort für Textformat einer Variablen, normalerweise zur Deklaration von Arrays. Mit Strings können keine mathematischen Operationen durchgeführt werden, jedoch können Strings addiert (verkettet) werden ("123" + "abc", ergibt "123abc").
Complex	Schlüsselwort für Zahlenformat einer Variablen (komplexe Zahl), normalerweise zur Deklaration von Arrays. Diese Zahlen bestehen aus einem Real- und einem Imaginär-Teil (a+bj). Beispiel: `x = -3+5j ; Komplexe Zahl`

List ( )	***List*** ist eine dynamische Datenstruktur. Alle Variablen-Typen (Zahlen, Strings, Kurven usw.) sind erlaubt. Auch gemischt. Sie kann unter Laufzeit erweitert oder gekürzt werden.
	
	Beispiel:
	`mylist = List() ; Liste deklarieren`
	`mylist() = 134  ; neues Element zufügen`
	`mylist() = "Text"`
	`mylist() = c0A1`
	
	`; Berechnungen durchführen. Resultat wird der Liste zugefügt`
	`mylist() = Smooth(mylist(2),20)`
	`tr1 = mylist(2) ; Original Kurve`
	`tr2 = mylist(3) ; geglättete Kurve`

Merge (arr*)	Verbindet mehrere Arrays zu einem. Dabei werden die einzelnen Arrays ***arr*** nacheinander verkettet, die einzelnen Werte innerhalb der Arrays bleiben unverändert. Die Arrays müssen vom selben Datentyp sein. **Beispiel:** `arr1 = array(1, 2, 3, 4)`

	`arr2 = array(5, 6, 7, 8)`
	`arr3 = array(9, 10, 11, 12)`
	
	`arr_new = Merge(arr1, arr2, arr3)`
	`; arr_new hat nun 12 Felder`

ZeroPadding (arr, NrOfZeros)	Fügt am Ende eines Arrays weitere Felder mit dem Wert 0 hinzu. ***arr*** ist eine Array, ***NrOfZeros*** die Anzahl zusätzlichen Felder im Array. Die bestehenden Felder werden nicht verändert, neue sind auf 0 gesetzt.
	
	Beispiel:
	`; Array mit 4 Feldern`
	`arr = array(1, 2, 3, 4)`
	`arr = ZeroPadding(arr, 10)`
	`; Array hat nun 14 Felder`

ConvToArray (trace)	Konvertiert ***trace*** zu einem Array. ***trace*** muss eine Kurve sein, der Rückgabewert ist ein Array. Zeit- und Trigger-Informationen gehen dabei verloren.
	
	Beispiel:
	`sig = c0A1`
	`arr = ConvToArray(sig)`

ConvToList (trace or array)	Konvertiert ***trace*** bzw. ***array*** in eine Liste. ***trace*** muss eine Kurve sein, ***array*** ein Array. Der Rückgabewert ist eine Liste.
	
	Beispiel:
	`ar$ = Array("A", "B", "C")`
	`CrsList = ConvToList(ar$)`

ConvToTrace (arr, samplefrq)	Konvertiert ***arr*** in eine Kurve. ***arr*** ist ein Array oder eine Liste vom Typ Double, ***samplefrq*** in Hz die Abtastrate der neuen Kurve. **1/*samplefrq*** ergibt den zeitlichen Abstand der einzelnen Samples in der neuen Kurve.
	
	Trigger Zeitpunkt wird bei 0s gesetzt, der Zeitstempel ist die aktuelle Uhrzeit des Computers zur Zeit der Konvertierung.
	
	Beispiel:
	`sf = 1E3 ;1kHz`
	`arr1 = array(1,2,3,4,5,6,7,8,9,10)`
	`trace = ConvToTrace(arr1, sf)`

Length (value)	***Length*** gibt die Grösse (Länge) von ***value*** zurück. *Value* kann einer der folgenden Datentypen sein:
	
	Array: Gibt die Anzahl Felder im Array zurück.
	List: Gibt die Anzahl Elemente in der Liste zurück.
	String: Gibt die Anzahl Zeichen im String zurück.
	Trace: Gibt die Anzahl der Samples in der Kurve zurück.
	
	Beispiel:
	`literal$ = "Test"`
	`len = Length(literal$)`
	`; len ist gleich 4`

26.10 Gruppe "Exponential and Trigonometric"

Exp (a)	Berechnet die **Exponentialfunktion** (e ^ a). ***a*** kann entweder eine Zahl oder eine Kurve sein und das Resultat wird wieder eine Zahl oder eine Kurve sein.
Ln (a)	Berechnet den **natürlichen Logarithmus** zur Basis e. ***a*** kann entweder eine Zahl oder eine Kurve sein und das Resultat wird wieder eine Zahl oder eine Kurve sein.
Log (a [,base])	Berechnet den **Logarithmus** zur Basis ***base***. ***a*** kann entweder eine Zahl oder eine Kurve sein und das Resultat wird wieder eine Zahl oder eine Kurve sein. **Wenn *base* nicht angegeben wird, wird der Logarithmus zur Basis 10 berechnet.**
Cos (a)	Berechnet den **Cosinus**. ***a*** kann entweder eine Zahl oder eine Kurve sein und das Resultat wird wieder eine Zahl oder eine Kurve sein. Die Werte von ***a*** werden als **Grad interpretiert** (Vollkreis = 360°).
Sin (a)	Berechnet den **Sinus**. ***a*** kann entweder eine Zahl oder eine Kurve sein und das Resultat wird wieder eine Zahl oder eine Kurve sein. Die Werte von ***a*** werden als **Grad interpretiert** (Vollkreis = 360°).
Tan (a)	Berechnet den **Tangens**. ***a*** kann entweder eine Zahl oder eine Kurve sein und das Resultat wird wieder eine Zahl oder eine Kurve sein. Die Werte von ***a*** werden als **Grad interpretiert** (Vollkreis = 360°).
Asin (a)	Berechnet den **Arkussinus**. ***a*** kann entweder eine Zahl oder eine Kurve sein. **Werte die ausserhalb +/-1 liegen werden als nicht definiert gekennzeichnet.** Das Resultat wird wieder eine Zahl oder eine Kurve sein. Ein Zahlen-Resultat wird in Grad (-90° bis +90°), ein Kurven-Resultat in Radiant (-Pi/2 bis +Pi/2°) angegeben.
Acos (a)	Berechnet den **Arkuscosinus**. ***a*** kann entweder eine Zahl oder eine Kurve sein. **Werte die ausserhalb +/-1 liegen werden als nicht definiert gekennzeichnet.** Das Resultat wird wieder eine Zahl oder eine Kurve sein. Ein Zahlen-Resultat wird in Grad (0° bis +180°), ein Kurven-Resultat in Radiant (0 bis Pi) angegeben.
Atan (a)	Berechnet den **Arkustangens**. ***a*** kann entweder eine Zahl oder eine Kurve sein und das Resultat wird wieder eine Zahl oder eine Kurve sein. Ein Zahlen-Resultat wird in Grad (-90° bis +90°), ein Kurven-Resultat in Radiant (-Pi/2 bis +Pi/2°) angegeben.
Atan (y, x)	Berechnet den **Arkustangens** von ***y*** **:** ***x*** entsprechend dem Vier-Quadranten-Polardiagramm. *x* und *y* können entweder beides Zahlen oder beides Kurven sein und das Resultat wird wieder eine Zahl oder eine Kurve sein. Ein Zahlen-Resultat wird in Grad (-180° bis +180°), ein Kurven-Resultat in Radiant (-Pi bis Pi) angegeben.

26.11 Gruppe "Spectrum (FFT)"

FFT (a [,FFTwindow])	Schnelle Fourier Transformation (Fast Fourier Transformation). Wird verwendet, um die Kurve ***a*** im Zeitbereich in ein Array mit komplexen Werten im Frequenzbereich zu wandeln.
	Es werden alle Werte von Beginn bis am Ende von ***a*** gewandelt. Allenfalls kann die Kurve mit der Funktion **Slice()** verkürzt werden.
	
	Der Parameter ***FFTwindow*** ist die Fensterfunktion zur Gewichtung des Signals im Zeitbereich.
	Wenn dieser fehlt, wird automatisch **RectangleWin** angenommen (d.h. alle Werte im Zeitbereich werden gleich gewichtet).
	Mit der Funktion **ConvToSpectrumTrace()** kann aus dem erzeugten Array eine Spektrum-Kurve gebildet werden, welche danach in ein Spektrum-Fenster (FFT Kurvenanzeige) gezogen werden kann.
	
	Beispiel:
	`tr = c0A1;`
	`arrc = FFT(tr, RectangleWin)`
	`tr_spec = ConvToSpectrumTrace(arrc)`

RectangleWin **HannWin** **HammingWin** **BlackmanWin** **FlatTopWin** **Taylor60Win** **Taylor80Win** **TriangleWin** **WelchWin** **BlackmanHarris3Win** **BlackmanHarris4Win**	Schlüsselwörter für den Parameter ***FFTwindow*** der Funktion ***FFT()***, Sie bestimmen das Bewertungsfenster beim Ausführen der Transformation.

ConvToSpectrumTrace (arrc)	Damit wird ein Array mit komplexen Werten ***arrc*** in eine Spektrum-Kurve gewandelt,
	Diese kann anschliessend in einem FFT-Kurvenanzeige-Fenster dargestellt werden.
	Die resultierende Spektrum-Kurve ist als Spitzenwert (Peak) skaliert.
	
	Beispiel:
	`tr = c0A1`
	`arrfft = FFT(tr, HannWin)`
	`trfft = ConvToSpectrumTrace(arrfft)`

IFFT (arrc)	Inverse Fourier Transformation. Damit kann ein Array aus komplexen Werten ***arrc*** im Frequenzbereich in ein Array im Zeitbereich gewandelt werden. Wenn eine Übereinstimmung mit der Original-Kurve vorausgesetzt wird muss die vorgängige **FFT()**-Berechnung mit der Rechteck-Gewichtung (**RectangleWin**) durchgeführt worden sein. **Beispiel:** `tr_orig = C0A1`

© Elsys AG

	``` ; parameters for later calculations sr = 1/TSample(tr_orig) Tb =TBegin(tr_orig) ; calculate FFT array arr_fft = FFT(tr_orig, RectangleWin) ; convert back to time array and a trace arr_time = ifft (arr_fft) tr_new = ConvToTrace(arr_time, sr) ; Adjust x axis tr_new = Shift(tr_new, Tb) ```

Angle (ac)	Berechnet den Winkelanteil von komplexen Zahlen.
	ac kann eine einzelne komplexe Zahl, oder ein Array aus komplexen Zahlen (z.B. Resultat von FFT()) sein. Das Resultat wird eine einzelne Zahl oder ein Array aus Double-Zahlen sein.
	Siehe auch **Abs(), Conjugate(), Real(), Imag().**
	
	Beispiel:
	`x = -3+5j ; Komplexe Zahl`
	`arc = angle(x) ; arc = -59.036°`

Conjugate (ac)	Mathematische Funktion für komplexe Zahlen. Invertiert den imaginären Anteil der einzelnen Elemente in ***ac***.
	ac kann eine einzelne komplexe Zahl, oder ein Array aus komplexen Zahlen (z.B. Resultat von FFT()) sein. Das Resultat wird wieder eine einzelne komplexe Zahl oder ein Array aus komplexen Zahlen sein.
	Siehe auch **Abs(), Angle(), Real(), Imag().**
	
	Beispiel:
	`x = -3+5j ;`
	`conx= Conjugate(x) ; conx = -3-5j`

Real (ac)	Gibt den/die realen Anteile von komplexen Zahlen zurück.
	ac kann eine einzelne komplexe Zahl, oder ein Array aus komplexen Zahlen (z.B. Resultat von FFT()) sein. Das Resultat wird wieder eine einzelne Zahl oder ein Array aus Double-Zahlen sein.
	Siehe auch **Abs(), Angle(), Conjugate(), Imag().**
	
	Beispiel:
	`x = -3+5j ;`
	`re = Real(x) ; re = -3`

Imag (ac)	Gibt den/die imaginären Anteile von komplexen Zahlen zurück.
	ac kann eine einzelne komplexe Zahl, oder ein Array aus komplexen Zahlen (z.B. Resultat von FFT()) sein. Das Resultat wird wieder eine einzelne Zahl oder ein Array aus Double-Zahlen sein.
	Siehe auch **Abs(), Angle(), Conjugate(), Real().**
	
	Beispiel:
	`x = -3+5j ;`
	`im = Imag(x) ; im = 5`

26.12 Gruppe "Report Generator"

Um auf die Funktionen in dieser Gruppe zugreifen zu können, muss zuerst via TranAX-Menu "Extras" die entsprechende Option installiert worden sein.

XlInitializeXML ( ) **XlInitializeCOM ( )**	**Einer der beiden Befehle muss ausgeführt werden bevor andere Report-Generator-Funktionen aufgerufen werden können**. Bei **XlInitializeXML()** braucht kein EXCEL vorhanden zu sein. Der erstellte Report wird durch eine XML-Datei (*.xlsx-Datei) gebildet. Diese kann danach auch durch EXCEL geladen werden. Bei **XlInitializeCOM() muss** eine gültige Version EXCEL ab 2007 installiert sein. Es wird eine Report-Mappe geöffnet, die auch direkt während oder nach der Messung mit EXCEL bearbeitet werden kann. Es empfiehlt sich Report-Einträge in vorbereitete Vorlagen (Templates) zu übertragen.

XlCreateEmptyFile ( )	Erstellt eine **leere** Excel Datei und gibt einen zufälligen Namen mit der Endung ".xlsx" als Rückgabewert zurück. Dieser Name muss weiterverwendet werden um auf der Datei arbeiten zu können.
	Per default ist dieser Datei der Pfad "data" des aktuellen Experiments zugewiesen.
	Diese Funktion empfiehlt sich, wenn errechnete Werte schnell in eine EXCEL-Datei gespeichert werden sollen. Sie beruht nicht auf einer zuvor erstellen Vorlage, kann aber im Nachhinein mit EXCEL weiterverarbeitet werden.
	Wenn diese Funktion aufgerufen wird, kann **XlOpenFile ()** entfallen.
	
	Beispiel:
	`fn$ = XlCreateEmptyFile()`
	`fn2$="MyRreport_#.xlsx" ; any own name`
	`XlSetCellValue(fn$, "A10", 7.123)`
	`XlSave(fn$, fn2$) ; fn2$ will be saved in ".\data"`

XlOpenFile (fname)	Öffnet eine bestehende Excel-Datei mit dem Namen ***fname*** (normalerweise eine Vorlage).
	Als Rückgabewert erhält man wieder den Datei-Namen ergänzt mit dem absoluten Datei-Pfad.
	Dieser muss in nachfolgenden Rerport-Generator-Funktionen verwendet werden.
	
	Hinweis: Wird Excel mit **XlInitializeXML()** initialisiert, können nur ***.xlsx** Dateien verwendet werden. Bei **XlInitializeCOM()** können auch ältere EXCEL-Dateiformate zur Anwendung kommen.
	
	Beispiel:
	`XlInitializeXML()`
	`fn$ = ".\Excel_Templates\Template_1.xlsx"`

	`filename$ = XlOpenFile(fn$)` `XlSetCellValue(filename$, "A10", 7)`

XlSelectSheet (fname, sheet)	Wählt in der EXCEL-Mappe ***fname*** das Tabellen-Blatt mit dem Name ***sheet***.
	Wenn dieser Befehl fehlt, wird standardmässig das erste Blatt gewählt.
	
	Beispiel:
	`fn$ = ".\Excel_Templates\Template_1.xlsx"`
	`filename$ = XlOpenFile(fn$)`
	`XlSelectSheet(filename$, "Tabelle2")`

XlGetCellValue (fname, cell)	Gibt den Wert in der Zelle ***cell***. zurück.
	In der Regel ist ***cell*** eine einzelne Zelle ("B3").
	Als ***cell*** wird ein String erwartet. Je nach dem was in der Zelle eingetragen ist, wird ein String oder eine Zahl zurück gegeben.
	Es kann aber auch ein Zellenbereich definiert werden (z.B. "A1:A4" oder "A1:D1"). Der Zellenbereich darf jeweils nur in derselben Kolonne oder Zeile definiert sein (nicht z.B. "A1:C3"). Auch muss vorher ein entsprechendes Array oder eine Liste deklariert worden sein. Es wird dann das Array oder die Liste abgefüllt mit den String- oder Zahlen-Werten der Zellen.
	Arrays müssen als String-Array oder Double-Array deklariert werden, je nachdem ob Strings oder Zahlen in den Zellen erwartet werden.
	Wenn eine Zelle leer ist, wird ein Leerstring bzw. "n. def." zurück gegeben. Wenn eine Liste deklariert wurde, enthalten danach deren Elemente Strings oder Zahlen.
	
	Beispiel:
	`fn$ = ".\Excel_Templates\Template_1.xlsx"`
	`fnam$ = XlOpenFile(fn$)`
	`v1 = XlGetCellValue(fnam$, "A1") ; Einzelwert`
	`valArr$ = Array(0 to 3) as String`
	`valArr$ = XlGetCellValue(fnam$, "A2:A5")`
	`valArrD = Array(0 to 3) as Double`
	`valArrD = XlGetCellValue(fnam$, "B2:E2")`
	`valArrL = List()`
	`valArrL = XlGetCellValue(fnam$, "A2:E2")`

XlGetCellReferences (fname, searchKey [, all])	Die Funktion sucht in der ganzen EXCEL-Mappe ***fname*** nach dem Schlüssel ***searchKey*** und gibt die gefundene Zellen-Bezeichnung zurück (z.B. "B5"). Diese Parameter müssen vom Typ String sein. Schlüsselwörter müssen natürlich vorher in der EXCEL-Mappe (Vorlage) eingetragen worden sein. Falls ***searchKey*** nicht gefunden wird, bringt die Funktion "**n. def."** zurück. Es empfiehlt sich Schlüsselwörter zu wählen, die sonst kaum vorkommen (z.B. "%V1"). Zwischen Klein- und Grossbuchstaben wird kein Unterschied gemacht. Falls der optionale Parameter ***all*** = **True** angegeben wird, wird ein Array bzw. eine Liste zurück gegeben. Darin sind alle Zellen-

	Bezeichnungen eingetragen, bei welchen das Schlüsselwort ***searchKey*** gefunden wurde.
	
	Beispiel:
	`fn$ = ".\Excel_Templates\Template_1.xlsx"`
	`fnam$ = XlOpenFile(fn$)`
	`cellRef = XlGetCellReferences(fnam, "%x")`

XlChangeColumn (cell, number)	Addiert die angegebene Zahl ***number*** zu der definierten Zelle ***cell***. Dabei wird die Kolonne von ***cell*** beinflusst.
	cell muss ein String (z.B. "B5"), ***number*** eine Zahl sein.
	
	Beispiel:
	`cell = "B2"`
	`newCell = XlChangeColumn(cell, 2) ; Ergibt "D2"`

XlChangeRow (cell, number)	Addiert die angegebene Zahl ***number*** zu der definierten Zelle ***cell***. Dabei wird die Zeile (row) von ***cell*** beinflusst.
	cell muss ein String (z.B. "B5"), ***number*** eine Zahl sein.
	
	Beispiel:
	`cell = "B2"`
	`newCell = XlChangeRow(cell, 2) ; Ergibt "B4"`

XlInsertImage (fname, winname, cell [, width [, height]])	Fügt den Bildschirminhalt im Fenster ***winname*** an der angegebenen Position ***cell*** in der EXCEL-Mappe **fname** ein.
	winname muss auch ein String sein. Dies kann der **Name** einer **einzelnen Kurvenanzeige**, einer **Skalar-Tabelle**, einer **ganze Seite** oder einer **Bild-Datei** (inkl. Pfadangaben) sein.
	cell ist in der Regel eine einzelne Zelle (z.B. "B2"). Sie bestimmt die linke obere Ecke des einzufügenden Bildes.
	Für ***cell*** kann aber auch ein Bereich angegeben werden (z.B. "A4:F30"). Der Zellen-Bereich bestimmt dabei den Ort und die Grösse des eingefügten Bildes.
	Fakultativ kann auch die Breite ***width*** und Höhe ***height*** des eingefügten Bildes definiert werden. Wird nur die Breite ***width*** angegeben, so wird die Höhe proportional dazu bestimmt (gleiche Seitenverhältnisse wie beim Original-Bild).
	width und ***height*** sind in Anzahl Pixels anzugeben.
	Wenn ein Zellen-Bereich (nicht nur eine Einzel-Zelle) vorgegeben ist, wird ***width*** und ***height*** ignoriert.
	
	Wenn unter "*Extras / Einstellungen / Benutzeroberfläche / Snapshot*" *Weisser Hintergrund* markiert ist, besitzt das übertragene Bild immer eine entsprechende Hintergrund-Farbe. Dabei sollten die Kurven-Farben eher dunkel gewählt werden.
	
	Beispiel:
	`fn$ = ".\Excel_Templates\Template_1.xlsx"`
	`fnam$ = XlOpenFile(fn$)`
	`XlInsertImage(fnam$, "Waveform 1", "A4:E20")`
	`XlInsertImage(fnam$, "Skalar_A 1", "A21",`
	`  100, 300)`

XlSetCellValue (fname, cell | searchKey, value)	Setzt in der EXCEL-Mappe ***fname*** den Wert ***value*** in die Zelle ***cell***. Anstelle einer Zelle (z.B. "B2") kann auch ein Schlüssel ***searchKey*** vorgegeben werden (dadurch kann **XlGetCellReferences** entfallen).
	fname und ***cell*** bzw. ***searchKey*** müssen Strings sein.
	value kann eine Zahl, String, Array, Liste oder Kurve sein.
	Wenn ***value*** ein Array, Liste oder eine Kurve ist, werden die Werte standardmässig in die **Kolonne** der spezifizierten Zelle ***cell*** eingefügt.
	Möchte man die Werte in eine **Zeile** einfügen, muss die Zelle folgendermassen definiert werden: **"A7:row".**
	
	Kurven (wie auch Arrays oder Listen) dürfen nicht zu lang sein für die Übertragung in EXCEL.
	Die maximale Länge (Anzahl Samplings, bzw. Elemente) beträgt:
	Übertragung in eine **Kolonne**: 1'048'577 - Linie des ersten Wertes (z.B. <=1'048'571, wenn ***cell*** = "x**6** ").
	Übertragung in eine **Zeile**: 16'385 - Kolonne des ersten Wertes(z.B. <=116'382, wenn ***cell*** = "**C**x:row ").
	Zu lange Kurven (grosse Blocklängen) können vorher mit **Slice(..)**, **Resampling(..)** oder **Skip(..)** verkürzt werden.
	
	Beispiel:
	`filename = "excel\Template.xlsx"`
	`filename = XlOpenFile(filename)`
	`colArr = Array(1,2,3,4)`
	`XlSetCellValue(filename, "A1", colArr)`
	`XlSetCellValue(filename, "A1:row", colArr)`

XlSetHeader (fname, alignment | searchKey, value)
 	
XlSetFooter (fname, alignment | searchKey, value)
	Setzt in der **Kopf-** bzw. **Fuss-Zeile** der EXCEL-Mappe ***fname*** den Wert ***value*** ein.
	Der zweite Parameter bestimmt, wo ***value*** eingefügt werden soll. Dieser kann ein Schlüsselwort **XlLeft, XlCenter, XlRight** (für Links, Zentriert, Rechts) oder ein Suchwort sein. Ein Suchwort muss vorher in der Kopf- bzw. Fuss-Zeile der EXCEL-Vorlage eingefügt worden sein.
	
	Beispiel:
	`Fn$ = "excel\Template.xlsx"`
	`Fn$ = XlOpenFile(Fn$)`
	`Skey$="%Hd-1" ; Should be set in Template`
	`XlSetHeader(Fn$, XlLeft, "Report")`
	`XlSetHeader(Fn$, Skey$, "Unit Test 1")`
	`XlSetFooter(Fn$, XlCenter, "Result Table")`

XlLeft **XlCenter** **XlRight**	Schlüsselwörter für den Parameter ***alignment*** der Funktionen ***XlSetHeader()*** bzw. ***XlSetFooter()***
XlSave (fname, fnameReport)	Speichert die offene EXCEL Datei ***fname*** inkl. allen Änderungen. Wenn die Vorlage nicht überschrieben werden soll, muss für den zweiten Parameter ***fnameReprot*** ein anderer Namen (inkl. Pfad) gewählt werden.

	Solange danach kein **CloseFile(*fname*)**-Befehl ausgeführt wird, bleibt die Excel-Mappe geöffnet. Es könnten also unmittelbar mit Excel zusätzliche manuelle Einträge gemacht werden.
	
	Beispiel:
	`XlInitializeXML ( )`
	`fn$ = ".\Excel_Templates\Template_1.xlsx"`
	`fnam$ = XlOpenFile(fn$)`
	`fnRep$ = ".\Excel_Reports\Report_#.xlsx"`
	
	`; Block with value insertions in EXECL Sheet.`
	
	`XlSave (fn$, fnRep$) ; Saves the EXCEL file`
	`(incl. all insertions) in the Report File.`

26.13 Gruppe "Recording Parameters"

(Funktionen noch nicht freigegeben)

26.14 Gruppe "Layout Waveforms"

SetXRange (WindowName, orientation, axisIndex, min, max)	Setzt das X-Achsen-Intervall der Kurvenanzeige ***WindowName*** auf die Werte ***min*** und ***max***.
	min und ***max*** können auch vertauscht sein, d.h. der kleinere Wert entspricht immer der linken Begrenzung.
	Die beiden Parameter ***orientation*** und ***axisIndex*** sind für zukünftige Erweiterungen reserviert.
	
	Beispiel:
	`; Sichtbarer Bereich der X-Achse in einem Zeit-Fenster ; wird von -2ms bis 5ms gesetzt`
	`SetXRange("Waveform 1", BottomAxis, 0, -2E-3, 5E-3)`

BottomAxis	Schlüsselwort, wird für den Parameter ***orientation*** in der Funktion ***SetXRange*** verwendet.

SetYRange (WindowName, areaIndex, orientation, axisIndex, min, max)	Setzt das Y-Achsen-Intervall des Bereiches ***areaIndex*** der Kurvenanzeige ***WindowName*** auf die Werte ***min*** und ***max***.
	Für den Parameter ***orientation*** verwendet man eines der beiden Schlüsselwörter ***LeftAxis*** oder ***RightAxis***.
	areaIndex bezeichnet den entsprechenden Bereich im Kurvenanzeige-Fenster, beginnend mit 0 von oben.
	axisIndex ist in den meisten Fällen 0. Sind mehrere Y-Achsen vorhanden, wird dieser Index entsprechenden von innen nach aussen inkrementiert.
	min und ***max*** können auch vertauscht sein, d. h. der kleinere Wert entspricht immer der unteren Begrenzung.
	Die Werte entsprechen den jeweiligen physikalischen Einheiten.
	
	Beispiel:
	`; Beispiel im obersten Bereich vom Kurvenanzeige-`
	`; Fenster Waveform 1. Zwei Y-Achsen auf der linken`
	`; und einer Achse auf der rechten Seite`
	`; 1. Achse links: -2 bis 5`
	`; 2. Achse links: -10 bis 20`
	`; 1. Achse rechts: -100 bis 200`
	
	`SetYRange("Waveform 1", 0, LeftAxis, 0, -2, 5)`
	`SetYRange("Waveform 1", 0, LeftAxis, 1, -10, 20)`
	`SetYRange("Waveform 1", 0, RightAxis, 0, -100, 200)`

LeftAxis **RightAxis**	Schlüsselwörter für den Parameter ***orientation*** in der Funktion ***SetYRange***.

26.15 Gruppe "Auto Sequence Functions"

StartRecording ( )	Startet eine Aufnahme.
StopRecording ( )	Bricht eine laufende Aufnahme ab.
WatOnEOR ( )	Wartet mit der weiteren Ausführung bis die Aufnahme beendet wurde (End Of Record). Falls im Control-Panel nicht *Single Shot* eingestellt ist, muss die Aufnahme manuell (oder durch den Befehl *Stop Recording*) abgebrochen werden um den EOR-Status zu erfüllen.
Trigger ( )	Löst einen Trigger aus (sofern dies nicht bereits durch das Signal selber geschehen ist).
AutoCalibration ( )	Führt eine Autokalibration aller Eingangskanäle durch.
EnableExternalStart ( )	Ermöglicht das Auslösen der Datenerfassung durch ein externes Signal am *Start Record* Eingang (normalerweise Pin 3 am 25pol. Stecker).
DisableExternalStart ( )	Verhindert das Auslösen der Datenerfassung durch ein externes Signal am *Start Record* Eingang (normalerweise Pin 3 am 25pol. Stecker).
Call (cmd, aguments [,TimeOut])	Dieser Befehl startet ein anderes Programm. Dies kann entweder eine .EXE oder eine .BAT Datei sein. ***TimeOut*** gibt an, wie lange auf die Beendigung des aufgerufenen Programms gewart werden soll. Wenn ***TimeOut*** fehlt, wird sofort mit dem nächsten Formel-Befehl weitergefahren. Das aufgerufene Programm läuft dann als neuer Prozess parallel zu TranAX. Wenn ***TimeOut*** grösser als Null angegeben wird, wartet die Autosequenz maximal so viele Sekunden auf die Beendigung des Programms. Wird das Programm vorher beendet, wird sofort mit dem nächsten Befehl weitergefahren. Ist ***TimeOut*** negativ, z.B. -1, wird ohne zeitliche Beschränkung auf die Beendigung des Programms gewartet.
Beep ( )	Der PC gibt einen Piepston von sich.
QuitApplication ( )	Das Messprogramm TranAX wird beendet.
Delay (seconds)	Verzögert den Ablauf entsprechend dem Parameter ***seconds***. Diese Funktion liefert keine genaue zeitliche Verzögerung und sollte deshalb **nicht für genaue Zeitmessungen eingesetzt werden!** Die Abweichungen hängen von parallel laufenden Prozessen im Betriebssystem ab. Diese können einige ms betragen.
WaitForCalculations ( )	Die Operation ***WaitForCalculations*** wartet bis die Berechnungen von **Skalar-** und/oder **Harmonischen-Tabellen** abgeschlossen sind.
LoadLayout (fn)	Lädt die Layout-Datei ***fn***.
LoadSettings (fn)	Lädt die Einstellungen-Datei ***fn*** ins Control Panel.
StoreSnapshot (fn, WiName)	Speichert das Kurvenanzegefenster ***WiName*** in die Datei ***fn***. Es werden die Einstellungen unter *"Extras / Einstellunge / Benutzeroberfläche ..."* berücklsichtigt. Bezüglich Datei-Pfad gilt das gleiche, wie unter "File(....)" beschrieben. Wenn ein "#" am Ende des Dateinamens ist, wird dieses Zeichen durch eine fortlaufende Nummer ersetzt.
SaveSpectrum (fn, WiName, Channel*)	Speichert das Spektrum einer FFT-Kurvenanzeige in die Datei ***fn*** (normalerweise ins Verzeichnis *"data"* des aktuellen Experiments).

	WiName bezeichnet das FFT-Spektrum Fensters. ***Channel****:* Alle Kanäle, von welchen Spektren in die Datei gespeichert werden sollen. Kanäle bei welchen kein Spektrum gerechnet wurde (bzw. berechnet werden konnte) werden nicht gespeichert. Die Spektren werden genauso gespeichert, wie sie in dem Spektrum-Fenster aufgrund der Zeitfensterbegrenzungen in der ursprünglichen Zeitkurve berechnet wurden. Auch die FFT-Gewichtung kann nachträglich nicht geändert werden. Nur Spektren von Hardware-Kanälen (z.B. c0A1) werden akzeptiert. Bezüglich Datei-Pfad gilt das gleiche, wie unter "File(....)" beschrieben. Wenn ein "#" am Ende des Dateinamens ist, wird dieses Zeichen durch eine fortlaufende Nummer ersetzt.
Print (WiName)	Druckt das Kurvenfenster ***WiName***. (Die angezeigten Kurven werden nach jeder Messung automatisch aktualisiert). Der Ausdruck erfolgt so wie unter *"Druckvorschau"* spezifiziert..
StorePage (fn, Page)	Speichert die Kurvenanzeige-Seite ***Page*** in die Datei ***fn*** (normalerweise ins Verzeichnis *"data"* des aktuellen Experiments). Bezüglich Datei-Pfad gilt das gleiche, wie unter "File(....)" beschrieben. Wenn ein "#" am Ende des Dateinamens ist, wird dieses Zeichen durch eine fortlaufende Nummer ersetzt.
StoreReadouts (Filename, Results*) **StoreReadouts (Filename, ScalarTableName, Line | Column, LineNr | ColNr [, Results]*)**	Speichert die durch Formeln ermittelten Einzelwert-Resultate ***Results*** in eine Text-Datei mit dem Namen ***Filename***. Speichert die Resultate aus der Skalar-Tabelle ***ScalarTableName*** in eine Text-Datei mit dem Namen ***Filename***. Mit den Parametern ***Line*** oder ***Column*** und der entsprechenden Nummer ***LinNr*** oder ***ColNr*** können die Zellen in der Skalar-Tabelle bestimmt werden. Optional können auch noch die durch Formeln ermittelten Einzelwert-Resultate ***Results*** angehängt werden. Bezüglich Datei-Pfad gilt das gleiche, wie unter "File(....)" beschrieben. Wenn ein "#" am Ende des Dateinamens ist, wird dieses Zeichen durch eine fortlaufende Nummer ersetzt.
Line	Schlüsselwort, welches für die Funktion **"StoreReadouts"** verwendet wird. Mit ***Line*** wird die entsprechende Linie/Zeile der Skalartabelle gespeichert. **Beispiel:** `StoreReadouts("readouts.txt", "Skalar_A 1", Line, 1)`
Column	Schlüsselwort, welches für die Funktion **"StoreReadouts"** verwendet wird. Mit ***Column*** bzw. ***Col*** wird die entsprechende Reihe/Spalte der Skalartabelle abgespeichert. **Beispiel:** `StoreReadouts("readouts.txt", "Skalar_A 1", Col, 1)`
IsRecording ( )	Überprüft, ob eine Messung am Laufen ist. Ist dies der Fall erhält man als Rückgabewert ***True*****.**

	Beispiel:
	`do while IsRecording()=True`
	`  StopRecording()`
	`loop`
	`StartRecording()`

WaitForData (TimeOut)	Wartet so lange bis mindestens ein gültiger Block aufgezeichnet wurde.
	Mit dem Parameter ***TimeOut*** kann eine maximale Warte-Zeit (in Sekunden) vorgegeben werden. Ist dieser Parameter negativ oder fehlt er, wird unendlich lang gewartet.
	Im Continuous-Mode wird nicht gewartet. Das gleiche gilt, wenn im ECR-Mode die Dual-Aufzeichnung aktiviert ist. Man verwende dazu die Funktion **WaitOnEOR**
	
	Beispiel:
	`StartRecording()`
	`WaitForData(100) ; wartet max. 100 sek.`
	`; Bem.: Es sind keine Daten vorhanden, wenn`
	`; TimeOut überschritten wurde.`

WaitOnEOR ( )	Wartet so lange bis die Messung beendet wurde (End Of Record).
	Die Aufzeichnung kann dabei auch durch einen Abbruch beendet worden sein. Normalerweise liegen dann keine gültigen Daten vor.
	
	Beispiel:
	`StartRecording()`
	`WaitOnEOR()`
	`; Bem.: Es sind nur Daten vorhanden, wenn`
	`; getriggert wurde.`

26.16 Gruppe "Signal Generations"

Mit den Funktionen in dieser Gruppe können durch reine mathematische Berechnungen Signal-Kurve erzeugt werden.

CreateSignal (NrOfPeriods, SampleRate, ValuePairs*)
 	
CreateSignal (NrOfPeriods, trace, ValuePairs*)
	Mit ***CreateSignal*** können beliebige Signal-Kurven erzeugt werden. ***NrOfPeriods*** gibt die Anzahl der Perioden (Wiederholungen) an. Dieser Wert wird jeweils auf eine ganze Zahl abgerundet und sollte somit >=1 sein. ***SampleRate*** definiert die virtuelle Abtastrate (in Hz) der erzeugten Kurve.
	Alternativ kann anstelle von ***SampleRate*** auch eine Kurve ***trace*** angegeben werden, dadurch werden die Einstellungen der X-Achse aus dieser übernommen.
	
	ValuePairs besteht immer aus zwei Zahlenwerten: Der erste für den Zeitpunkt auf der X-Achse, der zweite für den Amplitudenwert. Die Anzahl der Wertepaare ***ValuePairs*** ist nicht limitiert.
	
	Beispiele:
	`; Rechteck mit zehn Perioden`
	`rect = CreateSignal(10,2E6, 0,0, 0.005,0, 0.006,5, 0.011,5, 0.012,0, 0.017,0)`
	
	`; Amplituden-Schritt`
	`puls = CreateSignal(1,2E6, 0,-5 , 1e-3,-5 , 1e-3,5 , 10E-3,5)`

Sinus (ampl, freq, phase, samplerate, length)	Die Funktion gibt eine **sinusförmige Kurve** zurück. ***ampl***, ***freq*** (in Hz), ***phase*** (in °), ***samplerate*** (in Hz), und ***length*** (Anzahl Smaples) müssen Zahlen sein.
Sinus(ampl, freq, phase, trace)	Die Funktion gibt eine **sinusförmige Kurve** zurück. ***ampl, freq*** (in Hz) und ***phase*** (in °), müssen Zahlen sein. Das Resultat hat die gleichen zeitlichen Parameter (Abtastrate, Länge, Nullpunkt) wie ***trace***.
Noise (ampl, samplerate, length)	Die Funktion erzeugt ein **Rausch-Signal**. ***ampl, samplerate*** (in Hz), und ***length*** (in Samples) müssen Zahlen sein (Gauss-verteiltes Zufalls-Signal mit Standardabweichung ***ampl***).
Noise (ampl, trace)	Die Funktion erzeugt ein **Rausch-Signal**. ***ampl*** muss eine Zahl sein. (Gauss-verteiltes Zufalls-Signal mit Standardabweichung ***ampl***). Das Resultat hat die gleichen zeitlichen Parameter (Abtastrate, Länge, Nullpunkt) wie ***trace***.

Ramp (slope, samplerate, length)	Die Funktion erzeugt eine **Rampe** mit der Steilheit ***slope*** (Einheit pro Sekunde). ***slope***, ***samplerate*** (in Hz) und ***length*** (in Samples) müssen Zahlen sein.
	Beispiel:
	`sig[V] = ramp(3, 1E3, 10E3)`
	`; Rampe, 3V/s, 1kHz Abtastrate, 10kSample`

Ramp (slope, trace)	Die Funktion erzeugt eine Rampe mit der Steilheit ***slope*** (Einheit pro Sekunde). ***slope*** muss eine Zahl sein. Das Resultat hat die gleichen zeitlichen Parameter (Abtastrate, Länge, Nullpunkt) wie ***trace***.
Ramp(y0, y1, samplerate, length)	Die Funktion erzeugt eine **Rampe** mit der Steilheit aus ***y0*** und ***y1***. ***y0*** (in V oder Einheit), ***y1*** (in V oder Einheit), ***samplerate*** (in

	Hz) und ***length*** (in Samples) müssen Zahlen sein. Die Steilheit kann wie folgt festgelegt werden: $Steilheit\ \left[\frac{Einheit}{Sekunden}\right] = \frac{\Delta y}{\Delta x} = \frac{y1 - y0}{\frac{1}{samplerate} \times length}$
Ramp (y0, y1, trace)	Die Funktion erzeugt eine **Rampe** mit der Steilheit aus ***y0*** und ***y1***. ***y0*** *und* ***y1*** müssen Zahlen sein, ***trace*** eine Kurve. Die Steilheit wird aus ***y0***, ***y1*** und ***trace*** (Abtastrate und Blocklänge) bestimmt. Das Resultat hat die gleichen zeitlichen Parameter (Abtastrate, Länge, Nullpunkt) wie ***trace***.

26.17 Gruppe "Misc. Functions"

Diese Gruppe enthält diverse verschiedene Funktionen.

NBlks(trace)	Gibt die Anzahl der Blöcke einer Kurve wieder.
	
	Beispiele:
	`; sucht das Max. in allen Blöcken`
	`fname$ = "messung.tpc5"`
	`trace = file(fname$,0)`
	`blk = NBlks(trace)`
	`Mx=-1000  ; initialize`
	`i=0`
	`For  i = 0 to blk-1`
	`  value = Max(file(fname$,0).i)`
	`  If value > Mx Then`
	`   Mx = value`
	`  Endif`
	`Next`

PlaySound (trace or fname [, sr])	Audio- und tpc5 Dateien können über den Audio-Ausgang des Computers abgespielt werden. Unterstützt werden die Formate *.wav, *.mp3 und *.tpc5. Es können auch berechnete Kurven ***trace*** anstelle einer Datei abgespielt werden.
	
	Bei *.tpc5 Dateien bzw. ***trace*** muss der Parameter ***sr*** (Samplerate in Hz) angegeben werden. Die maximale Samplerate, wie auch die maximale Amplitude, hängt vom installierten Audio System ab. D.h. ein Abspielen direkt von einer *.tpc5 Datei ist nur dann gewährleistet, wenn die Kurve auf der Datei eine entsprechende Abtastrate aufweist. Es empfiehlt sich, durch ein vorangehendes Resampling mit Skalierung die Kurve auf eine geeignete Audio-Frequenz und Amplitude zu konvertieren, siehe Beispiel.
	
	Bei *.wav und *.mp3 fehlt der Parameter ***sr***, diese Information wird direkt aus der Datei gelesen.
	
	Beispiel:
	`AmpFact=0.5`
	`; Linker Kanal`
	`sLeft=File("SoundData.tpc5",0)`
	`; Rechter Kanal`
	`sRight=File("SoundData.tpc5",1)`
	`; Anpassen an Audio-Freq.`
	`sLeft=AmpFact*Resampling(sLeft, 44100)`
	`sRight=AmpFact*Resampling(sRight, 44100)`
	
	`; Linker u. rechter Kanal in einer Datei`
	`Save ("SoundStereo.tpc5", sLeft, sRight)`
	
	`srRec=1/TSample(sLeft)`
	
	`; Abspielen`
	`PlaySound("SoundStereo.tpc5", srRec)`

Audio_Sr_44100	Liefert den Wert 44100. Diese Konstante kann in der Funktion **PlaySound()** verwendet werden um die Standard-Samplerate anzugeben.

GetNrOfDevices ( )	Gibt die Anzahl der TraNET-Geräte, die via Sync.Link in einem System sind an. Normalerweise ist nur ein Gerät in einem System. Mehrere Geräte müssen zusätzlich zum Sync.Link via Menü ***"Datei" / "Gerät neu wählen"*** gemeinsam verbunden werden.
GetNrOfBoards (device)	Gibt die Anzahl **Boards** (Module) im Gerät ***device*** an. ***device* = 0** entspricht dem ersten Gerät.
GetNrOfInputs (device, board)	Gibt die Anzahl **Kanäle** auf einem ***board*** (Modul) im Gerät ***device*** an. ***device* = 0** entspricht dem ersten Gerät. ***board* = 0** entspricht dem ersten Modul "A" im Gerät.

SetFormulaError (msg)	***msg*** ist ein String. Die Funktion kann z.B. dazu verwendet werden um an einer bestimmten Stelle eine Fehlermeldung abzusetzen und dadurch zu kennzeichnen, dass weitere Funktionen nicht mehr gültige Resultate liefern werden
	
	Beispiel:
	`Tb=TBegin(trc)  ; initialize`
	`for a = 1 to 10`
	`  Tcr = TCross(trc,tb,TEnd(trc),0)`
	`  tb = Tcr ; for next`
	`  If Tcr = NotDefined Then`
	`    text$ = "Zu wenig Nulldurchgänge"`
	`    SetFormulaError(text$)`
	`  endif`
	`next`

MergeTraces ([SeperationMark,] Trace*)	Fügt mehrere Kurven zu einer einzigen zusammen.
	Wenn der fakultativen Parameter ***SeperationMark*** =**True** gesetzt wird, werden im zusammengefügten Kurvenzug zwischen den einzelnen Original-Kurven spezielle Marken erzeugt.
	
	Beispiel:
	`s10=c0A1.0 ; Block 0 (Multiblock Recording)`
	`s11=c0A1.1 ; Block 1`
	`s12=c0A1.2 ; Block 2`
	
	`s1All=MergeTraces(True, s10, s11, s12)`

UseMemory (bool)	Falls ***bool*** = *True* ist, werden Kurven-Resultate (Traces) im Arbeitsspeicher und nicht als Datei im Verzeichnis ".\expr\" abgelegt. Dies kann die Berechnungszeit erheblich verkürzen. **UseMemory(bool)** entspricht der Funktion *SetEnvironment(0, True).*
Pi	Die Kreiszahl ***Pi***. π = 3.14159...

GetVersion (id [, deviceIdx [, boardIdx]])

Versionsnummer von Hard- oder Software abfragen.
Je nach verwendetem Schlüsselwort für ***id*** werden weitere Parameter ***deviceIdx*** und ***boardIdx*** benötigt.

Id	deviceIdx	boardIdx
ProgramVersion		
ServerVersion	✓	
BoardDriverVersion	✓	✓
BoardFirmwareVersion	✓	✓
BoardHardwareVersion	✓	✓
InputHardwareVersion	✓	✓

Die Funktion gibt einen String zurück.

Beispiele:

```
; Verwendete TranAX Version
 ProgVer$=GetVersion(ProgramVersion)
 ; Strip last 4 digits
 ProgVer$=Slice(ProgVer$, 0, Length(ProgVer$)-6)
 If ProgVer$<> "3.4.0" then
  SetFormulaError("Software does not match")
 endif

; TPCServer Version von erstem Device
 ServVer$=GetVersion(ServerVersion,0)
; Treibervers. vom ersten Board im 1. Device
 DrvVer00$=GetVersion(BoardDriverVersion,0,0)
; Firmwarevers. vom ersten Board im 1. Device
 FWver00$=GetVerson(BoardFirmwareVersion,0,0)
; Hardware-Vers. der Grundkarte vom ersten
; Board im Ersten Device
 HWver$=GetVersion(BoardHardwareVersion,0,0)
; Version des Verstärkers vom ersten Board im
; Ersten Device
AmpVer$=GetVersion(InputHardwareVersion,0,0)
```

ProgramVersion
ServerVersion
BoardDriverVersion
BoardFirmwareVersion
BoardHardwareVersion
InputHardwareVersion

Schlüsselwörter zu der Funktion **GetVersion**.

27 Liste der Auto Sequenz Befehle

Autocalibration	Führt eine Autokalibration aller Eingangskanäle durch.
Beep	Wenn dieser Befehl ausgeführt wird, gibt der PC einen Piepston von sich.
Calculate	Dieser Befehl führt die im Formeleditor aufgeführten Kurvenberechnungen durch (siehe auch *Load Formula*).
Call	Dieser Befehl startet ein anderes Programm. Dies kann entweder eine .EXE oder eine .BAT Datei sein. Die Anführungszeichen sind zwingend notwendig und enthalten die Kommandozeile, welche auch Parameter für das aufgerufene Programm enthalten kann. Der Wartezeit-Wert gibt an, ob und wie lange die Autosequenz auf die Beendigung des aufgerufenen Programms warten soll. Wenn 0 angegeben wird, wartet die Autosequenz nicht, sondern fährt sofort mit dem nächsten Befehl weiter. Das aufgerufene Programm läuft dann als neuer Prozess parallel zu TranAX. Wenn eine Zahl grösser als Null angegeben wird, wartet die Autosequenz maximal so viele Sekunden auf die Beendigung des Programms. Wird das Programm vorher beendet, macht die Autosequenz sofort mit dem nächsten Befehl weiter. Wird eine negative Zahl angegeben, z.B. -1, so wartet die Autosequenz ohne zeitliche Beschränkung auf die Beendigung des Programms.
Comment	Hinzufügen eines Kommentars zur Autosequenz.
Delay	Wartet die Anzahl angegebener Sekunden (auch Bruchteile von Sekunden möglich, z.B. 1.5).
Disable External Start	Verhindert das Auslösen der Datenerfassung durch ein externes Signal am *Start Record* Eingang (normalerweise Pin 3 am 25pol. Stecker).
Enable External Start	Ermöglicht das Auslösen der Datenerfassung durch ein externes Signal am *Start Record* Eingang (normalerweise Pin 3 am 25pol. Stecker)..
Load Formula	Lädt die angegebene Formel-Datei zur allfälligen Kurvenberechnung via *Calculate*.
Load Layout	Lädt die angegebene Layout-Datei.
Load Settings	Lädt die angegebene Einstellungen-Datei ins Control Panel.
Next	Dieser Befehl ist der Schleifenrücksprung-Punkt zu *Repeat*. Falls *Repeat* nicht in der Autosequenz vorhanden ist, springt die Autosequenz zum ersten Befehl.

Pause	Hält die Autosequenz an. Der Benutzer muss die Fortsetzung bestätigen. Dieser Befehl kann verwendet werden, um Meldungen wie z.B. "Bitte Prüfling aktivieren" auszugeben, oder dem Benutzer die Möglichkeit zu geben, eine abgeschlossene Messung anzuschauen, bevor die Autosequenz fortgeführt wird.
Print	Druckt die aktuell dargestellten Kurven aus. (Die angezeigten Kurven werden nach jeder Messung automatisch aktualisiert). Der Ausdruck erfolgt so wie unter "Druckvorschau" spezifiziert. **Achtung:** es wird das Kurvenfenster gedruckt das aktuell den Fokus hat.
Quit Application	Dieser Befehl bewirkt, dass das Messprogramm TranAX verlassen wird.
Repeat	Stellt den Startpunkt einer Schleife dar. Dieses Label wird mit *Next* verwendet um eine Reihe von Befehlen mehrmals innerhalb einer Schleife auszuführen. Schleifen können auch verschachtelt werden,
Save	Speichert die angegebenen Kanäle. **Save <Dateiname>, <Kurve>, <Kurve>, ...** *Dateiname:* Definiert den Namen der Datei, in die die Daten gespeichert werden. Wenn ein "#" am Ende des Dateinamens ist, wird dieses durch eine fortlaufende Nummer ersetzt, z.B. "CHECK#.tpc5" "CHECK001.tpc5, CHECK002.tpc5, ...". *Kurve:* Via Formel-Editor berechnete Kurve, nur Name wie z.B xx oder abc. *Erfasste Kurve:* Kanalbezeichner mit optionaler Angabe des Blockes oder Blockbereiches, z.B. 0A1, 0B2[1], 0B4[1-10] *Kanalbezeichner:* <device><board><input>, z.B. 0A1 mit <device>=0, <board>=A, <input>=1 *Blockbereich:*[<blockbereich>] oder [<blockbereich>:<dataRed>] *dataRed:* Datenreduktion ist optional und nur zulässig für erfasste Kurven, nicht zulässig für berechnete Kurven. Die Optionen sind: *None:* keine Datenreduktion *Skipn:* Überspringt die Anzahl n-Werte. "Skip5" speichert jeden fünften Wert. *Averagen:* Bildet den Mittelwert über n-Werte in einem Punkt ab. *Average10* mittelt 10 Werte und bildet das in einen Wert ab. *MinMaxn:* Berechnet zwischen den n-Werten die Maximalwerte und bildet die neue Kurve alternierend aus Maxima und Minima. Bei Autosequenzen in autonom laufenden Geräten sind, mit Ausnahme der fortlaufenden Nummerierung des Dateinamens und des Blockbereichs, keine zusätzlichen Varianten möglich. Es werden immer TPC5 Dateien erzeugt. Die Endung

	*.tpc5 wird automatisch angehängt.
	
	Beispiele:
	`Save xx.tpc5, 0A1, 0A2[1], 0A2[2-10:Average10], xx, abc, 1B1-4[1-5]`
	`Save c:\data\Nr_#.tpc5, 0A1-4[3-5:Skip100]`
	`Save ref1, 0A1-4`

Save Spectrum	Speichert das Spektrum einer FFT-Kurvenanzeige.
	
	Save Spectrum (*.tps5) <Filename>, <Spectrum Window>, <Channels>
	
	Filename: Der gewünschte Dateiname (Die Datei gelangt in das Datenverzeichnisses des aktuellen Experiments)
	Spectrum Window: Bezeichner des FFT-Spektrum Fensters.
	Channels: Alle Kanäle, von welchen Spektren in die Datei gespeichert werden sollen. Kanäle bei welchen kein Spektrum gerechnet wurde (bzw. berechnet werden konnte) werden nicht gespeichert. Die Spektren werden genauso gespeichert, wie sie in dem Spektrum-Fenster aufgrund der Zeitfensterbegrenzungen in der ursprünglichen Zeitkurve berechnet wurden. Auch die FFT-Gewichtung kann nachträglich nicht geändert werden.
	Nur Spektren von Hardware-Kanälen (z.B. 0A1) werden akzeptiert.
	
	Beispiele für "Save Spectrum":
	`Save Spectrum (*.tps5) FilterSpec.tps5, Spectrum 1, 0A1`
	`Save Spectrum (*.tps5) FilterSpec_#.tps5, Spectrum 1, 0A1-4, 0B1`

Store Page	Speichert die Kurvenanzeige-Seiten (Pages).
	
	Store Page <Filename>, <Page name>
	
	Filename: Der gewünschte Dateiname (Die Datei gelangt in das Data-Verzeichnisses des aktuellen Experiments)
	Page Name: Hier gibt man an, welche Kurvenanzeige-Seite gespeichert werden soll
	
	Beispiel für "Store Page":
	`Store Page HighVoltageExp.tdp, Seite 1`
	`Store Page HighVoltageExp_#.tdp, Page-3-Phases`

Start Recording	Startet eine Aufnahme.
Stop Recording	Bricht eine laufende Aufnahme ab.

Store Readouts	Speichert Resultate aus einer Skalarfunktions-Tabelle in eine Datei.
	Store Readouts <Fenstername>, "<Filename>", Lin or **Col nr, [fres1]**
	
	Fenstername: Name der Skalar-Tabelle (es kann dies eine Sakalar-Tabelle A oder B sein).
	Filename: Der gewünschte Dateiname (die Datei gelangt in das Data-Verzeichnis des aktuellen Experiments)
	Lin oder Col: Gibt an, ob die Werte aus einer Linie oder Kolonne gespeichert werden sollen.
	nr : Nummer der Linie bzw. der Kolonne
	fres1: Falls danach noch "fres1" angegeben wird, werden auch die Resultate der Einzelwert-Berechnungen aus dem Formel-Editor abgespeichert (dieser Parameter muss genau *fres1* heissen).
	Hinter den Werten werden auch die Einheiten (falls vorhanden) abgespeichert (diese können ja in den Tabellen ein- oder ausgeschaltet werden).
	
	Eine Readout-Datei wird beim ersten ***Store Readouts*** Aufruf in der Autosequenz neu gebildet, bzw. überschrieben. Wenn ein "#" am Ende des Dateinamens ist, wird dieses durch eine fortlaufende Nummer ersetzt.
	Vor dem ersten Eintrag von Skalar-Werten, werden die Titel der Linien bzw. Kolonnen in einer übergeordneten Text-Linie in die Datei geschrieben.
	Bei darauf folgenden ***Store Readouts*** Befehlen in die gleiche Datei (z.B. in einem Loop), werden die entsprechenden Skalar-Werte in neuen Text-Linien in der Datei unten abgelegt. Man erhält dadurch eine geeignete tabellarische Darstellung beim Aufzeichnen einer Messreihe.
	Die einzelnen Werte werden in der Text-Datei mit Tabulator getrennt.
	
	Beispiele für *Store Readouts*:
	`Store Readouts Skalar_A 1,Messwerte1-#.txt, Col2`
	`Store Readouts Skalar_B 2, Messwerte2-#.txt, Line1, fres1`

Store Snapshot	Die "Store Snapshot" Funktion kann in der **Auto Sequenz** verwendet werden um einen **Screenshot** der aktuellen **Kurvenanzeige-Fenster oder Tabelle** zu speichern. **Store Snapshot <Filename>, <Fenstername>** *Filename:* Der gewünschte Dateiname (ohne Pfadangabe gelangt die Datei in das Data-Verzeichnis des aktuellen Experiments). Wenn ein "#" am Ende des Dateinamens ist, wird dieses Zeichen durch eine fortlaufende Nummer ersetzt.

	Fenstername: Name des Fensters, von welchem ein Screenshot gespeichert werden soll. Es kann ein Waveform/Kurvenanzeige-Fenster (Zeit oder Spektrum) oder eine Tabelle (Skalar-Tabelle A, B oder Harmonische) deklariert werden.
	
	Beispiele für "Store Snapshot":
	`Store Snapshot ImgSpec1_#, Spektrum 1`
	`Store Snapshot ImgTabA1_#, Skalar_A 1`
	
	 Eine Kurvenanzeige wird anhand der Einstellungen für "Snapshot" im Menü ***"Extras" / "Einstellungen" / "Benutzeroberfläche"*** entweder als *.emf oder *.bmp abgelegt.
	
	 Skalartabellen werden unabhängig der Einstellungen immer als *.bmp gespeichert.

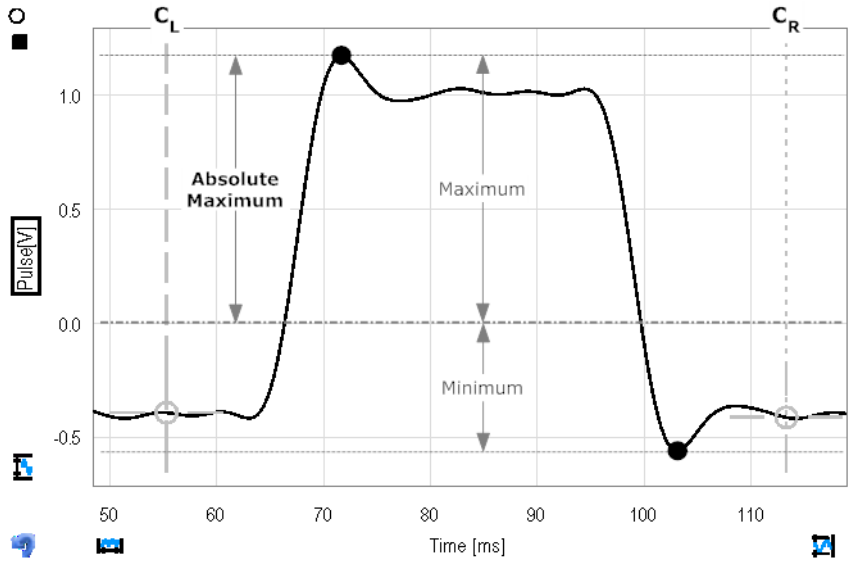
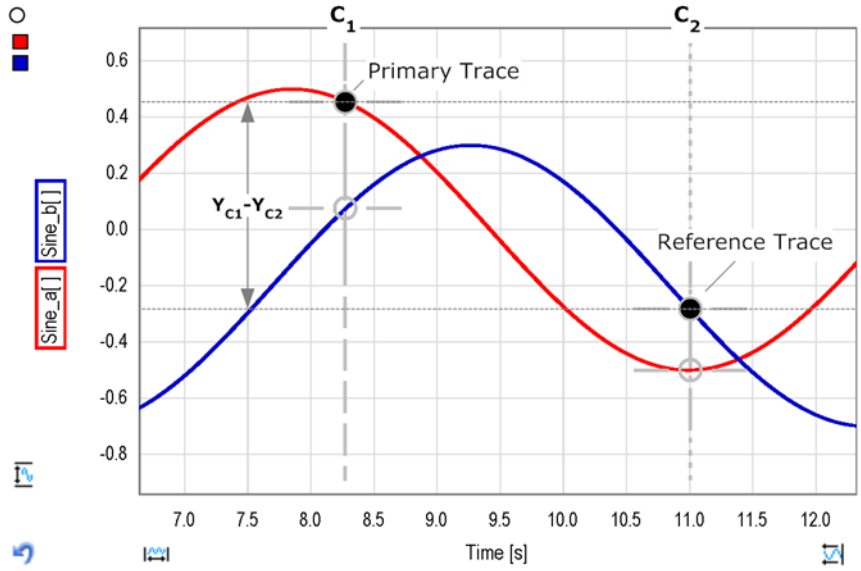
Trigger	Löst einen Trigger aus (sofern dies nicht bereits durch das Signal selber geschehen ist).
Wait for Calculations	Wartet mit der weiteren Ausführung der Autosequenz bis die Berechnungen der Formeln und Skalarfunktionen beendet wurden.
Wait on EOR	Wartet mit der weiteren Ausführung der Autosequenz bis die Aufnahme beendet wurde (End Of Record). Falls im Control-Panel nicht *Single Shot* eingestellt ist, muss die Aufnahme manuell (oder durch den Befehl *Stop Recording*) abgebrochen werden um den EOR-Status zu erfüllen

28 Liste der Skalarfunktionen

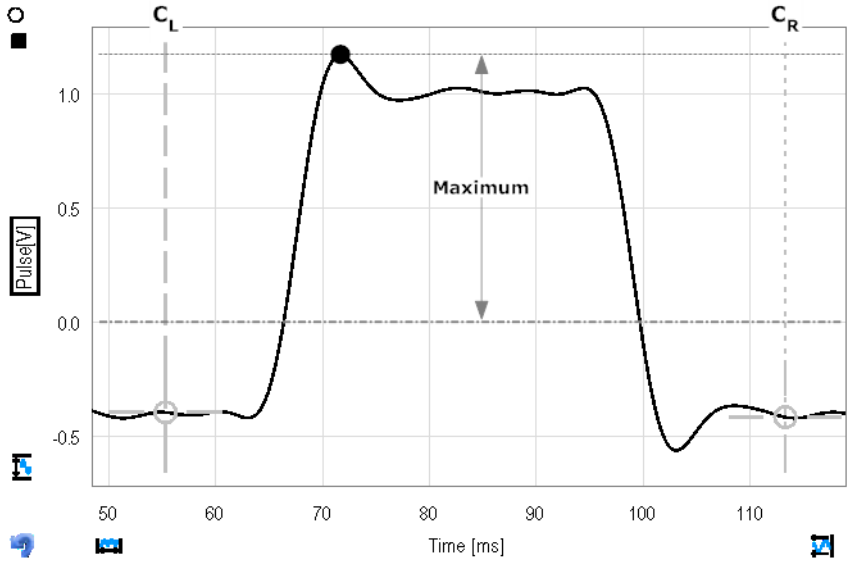
28.1 Gruppe "All Functions"

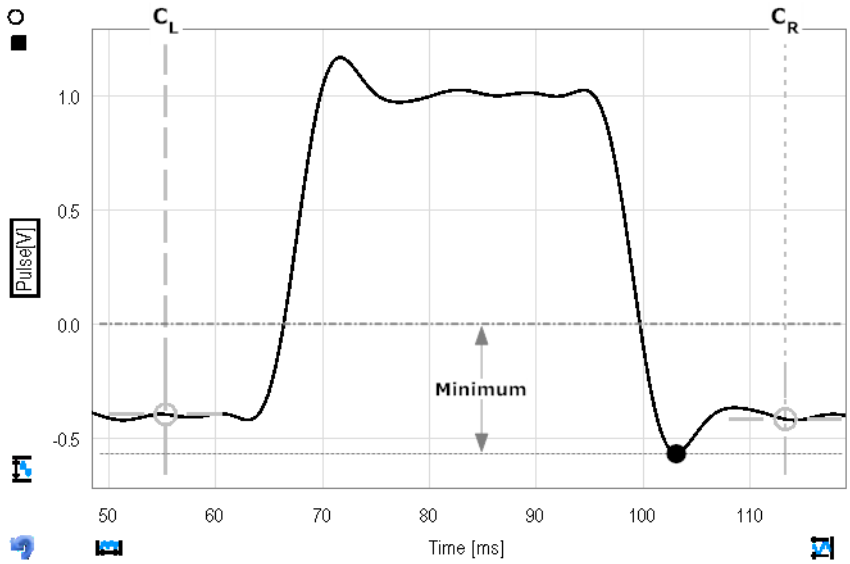
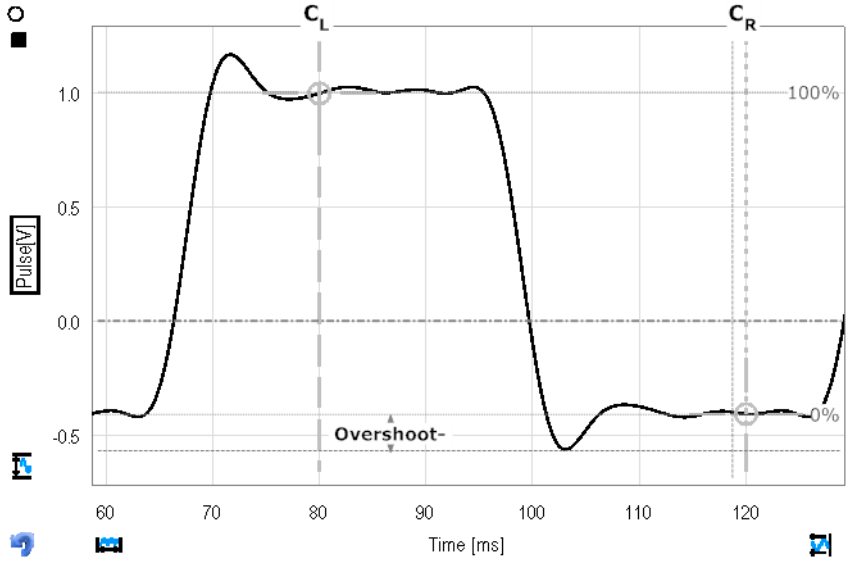
Cursor Ratio Amplitude (dB)	Berechnet das Amplitudenverthältnis zwischen Cursor 1 und Cursor 2 und skaliert dieses in dB. Diese Skalarfunktion ist **nur für Spektren (FFT)** verfügbar. Für Peak, RMS und Phase (Einstellung der Y-Skala) gilt: $Cursor\ Ratio\ [dB] = \ 20 * Log_{10}\left(\frac{Cursor_1}{Cursor_2}\right)$ Für Power und RMS^2 (Einstellung der Y-Skala) gilt: $Cursor\ Ration[dB] = \ 10 * Log_{10}\left(\frac{Cursor_1}{Cursor_2}\right)$
Frequency at Maximum	Sucht das Maximum zwischen Cursor A und Cursor B und gibt die Frequenz an der Position des Maximums zurück. Diese Skalarfunktion ist **nur für Spektren (FFT)** verfügbar.

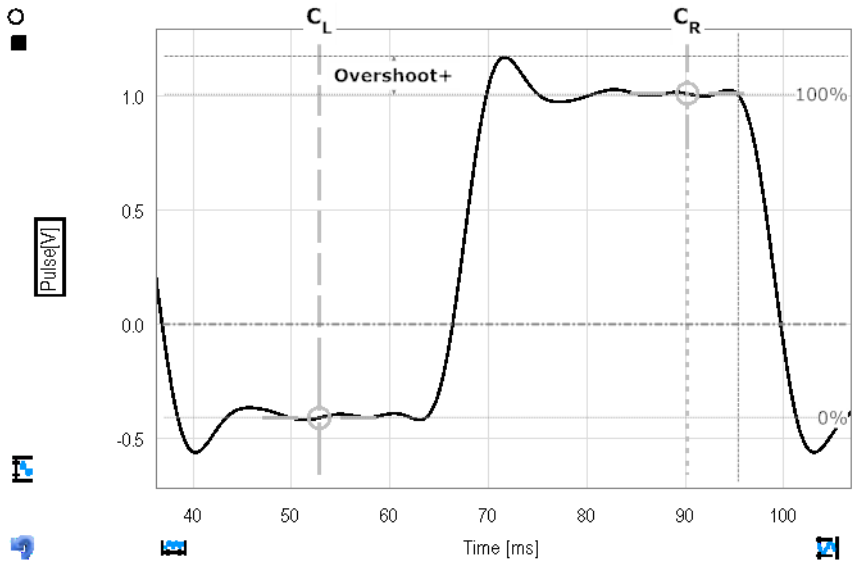
28.2 Gruppe "Vertical"

Absolute Maximum	Sucht den grössten absoluten Kurvenwert. Die Funktion liefert den kleinsten, negativen Wert, wenn dessen Betrag grösser ist als der grösste positive Wert.  $Absolute\ Maximum = \lvert Max\rvert$ if $\lvert Max\rvert \geq \lvert Min\rvert$ *or* $= Min$ if $\lvert Min\rvert > \lvert Max\rvert$ C_L Cursor left C_R Cursor right Max Maximum Min Minimum
Delta	Ermittelt die Amplituden-Differenz zweier Kurven zwischen zwei Cursors. Die Delta-Funktion erlaubt die Kombination von beliebigen Cursor-Paaren mit zwei unabhängigen Kurven.  $Delta = y_{C1\ Primary\ Trace} - y_{C2\ Reference\ Trace}$ C_1 Selected Cursor 1 C_2 Selected Cursor 2

 © Elsys AG

Maximum	Sucht den grössten Punkt in der Kurve relativ zur Signal-Nulllinie.  C_L Cursor left C_R Cursor right
Mean	Berechnet den Mittelwert des Signals. $$Mean = \frac{1}{n \cdot \Delta t} \cdot \sum_{i=C_L}^{C_R} y_i \cdot \Delta t$$ C_L Cursor left C_R Cursor right n # of Samples y_i y-value at position i Δt Sampling Interval
Mean Periodic	Sucht die Basislinie-Durchgänge des Signals und berechnet den Mittelwert nur für die gefundenen ganzen Perioden. $$Mean_{Periodic} = \frac{1}{n \cdot \Delta t} \cdot \sum_{i=LC_{Left}}^{LC_{Right}} y_i \cdot \Delta t$$ LC_L Level crossing left LC_R Level crossing right n # of Samples y_i y-value at position i Δt Sampling Interval
Minimum	Sucht den kleinsten Punkt in der Kurve relativ zur Signal-Nulllinie.

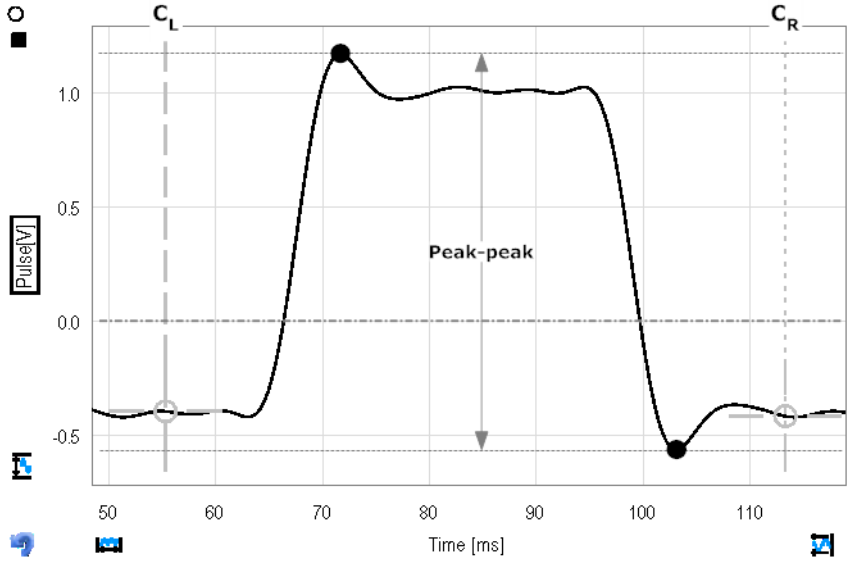
	 C_L Cursor left C_R Cursor right
Overshoot -	Gibt die Stärke der Unterschwingung zurück. Der linke Cursor soll auf die Puls-Schulter gelegt werden (100%) und der rechte Cursor auf die Puls-Grundlinie (0%).  C_L Cursor left (Top at 100%) C_R Cursor right (Base at 0%) Overshoot- Negative overshoot [%]
Overshoot +	Gibt die Stärke der Überschwingung zurück. Der linke Cursor soll auf die Puls-Grundlinie (0%) und der rechte Cursor auf die Puls-Schulter (100%) gelegt werden.



C_L	Cursor left (Base at 0%)
C_R	Cursor right (Top at 100%)
Overshoot+	Positive overshoot [%]

Peak-Peak

Berechnet die Differenz von Maximum und Minimum.



$$Peak\text{-}Peak = maximum - minimum$$

C_L	Cursor left
C_R	Cursor right

Rectified Mean

Berechnet den gleichgerichteten Mittelwert des Signals.

$$Rect = \frac{1}{n \cdot \Delta t} \cdot \sum_{i=C_L}^{C_R} |y_i| \cdot \Delta t$$

C_L	Cursor left
C_R	Cursor right
n	# of Samples

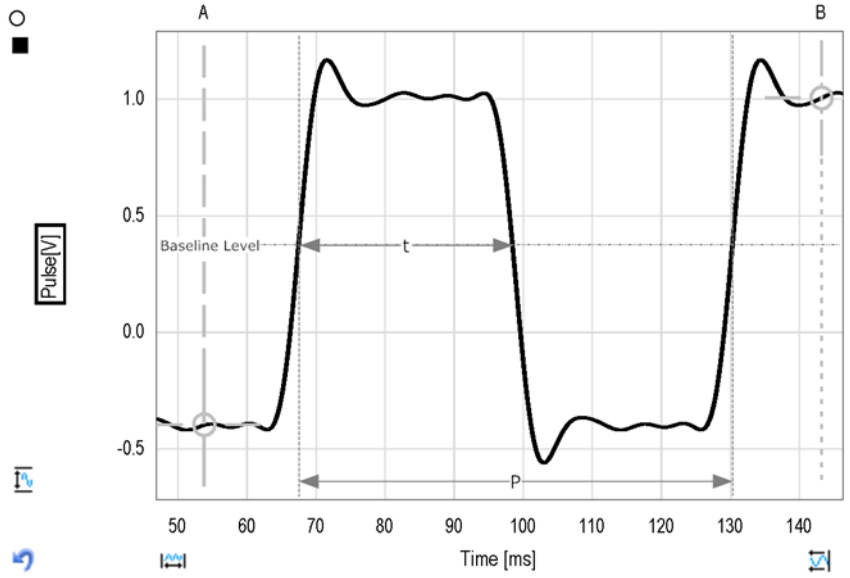
	y_i y-value at position i Δt Sampling Interval
Rectified Mean Periodic	Berechnet den gleichgerichteten Mittelwert des Signals von ganzen Perioden. Die Basislinie definiert Anfang und Ende von ganzen Perioden. $$Rect_{Periodic} = \frac{1}{n \cdot \Delta t} \cdot \sum_{i=LC_L}^{LC_R} \lvert y_i \rvert \cdot \Delta t$$ LC_L Level Crossing left LC_R Level Crossing right n # of Samples y_i y-value at position i Δt Sampling Interval
Rightmost Value	Zeigt den Amplituden-Wert der aktuellen Kurve am rechten Rand des entsprechenden Kurvenfensters an. Diese Funktion eignet sich, um im kontinuierlichen Mode den aktuellen Messwert anzuzeigen während die Aufnahme läuft. Diese Messgrösse ist Cursor-unabhängig. Diese Messgrösse wird auch bei laufender Erfassung angezeigt.
RMS	Berechnet den quadratischen Mittelwert (Effektivwert). $$RMS = \sqrt{\frac{1}{n \cdot \Delta t} \cdot \sum_{i=C_L}^{C_R} y_i^2 \cdot \Delta t}$$ C_L Cursor left C_R Cursor right n # of Samples y_i y-value at position i Δt Sampling Interval
RMS Periodic	Berechnet den Effektivwert der ganzen Perioden. Die Basislinie definiert Anfang und Ende von ganzen Perioden. $$RMS_{Periodic} = \sqrt{\frac{1}{n \cdot \Delta t} \cdot \sum_{i=LC_L}^{LC_R} y_i^2 \cdot \Delta t}$$ LC_L Level crossing left LC_R Level crossing right n # of Samples

	y_i y-value at position i Δt Sampling Interval
Standard Deviation	Berechnet die Standard-Abweichung des Signals. Die Standard-Abweichung entspricht einer RMS-Berechnung mit unterdrückter Nullpunkt-Verschiebung. $$SDev = \sqrt{\frac{1}{n \cdot \Delta t} \cdot \sum_{i=C_L}^{C_R} (y_i - mean)^2 \cdot \Delta t}$$ C_L Cursor left C_R Cursor right n # of Samples y_i y-value at position i Δt Sampling Interval
Standard Deviation Periodic	Berechnet die Standard-Abweichung des Signals von ganzen Perioden. Die Basislinie definiert Anfang und Ende von ganzen Perioden. Die periodische Standard-Abweichung entspricht einer periodischen RMS-Berechnung mit unterdrückter Nullpunkt-Verschiebung. $$SDev_{Periodic} = \sqrt{\frac{1}{n \cdot \Delta t} \cdot \sum_{i=LC_L}^{LC_R} (y_i - mean)^2 \cdot \Delta t}$$ LC_L Level Crossing left LC_R Level Crossing right n # of Samples y_i y-value at position i Δt Sampling Interval

28.3 Gruppe "Horizontal"

Duty Cycle

Duty Cycle gibt das Verhältnis zwischen positiver Pulsbreite und Perioden-Dauer beim Basislinien-Pegel an.

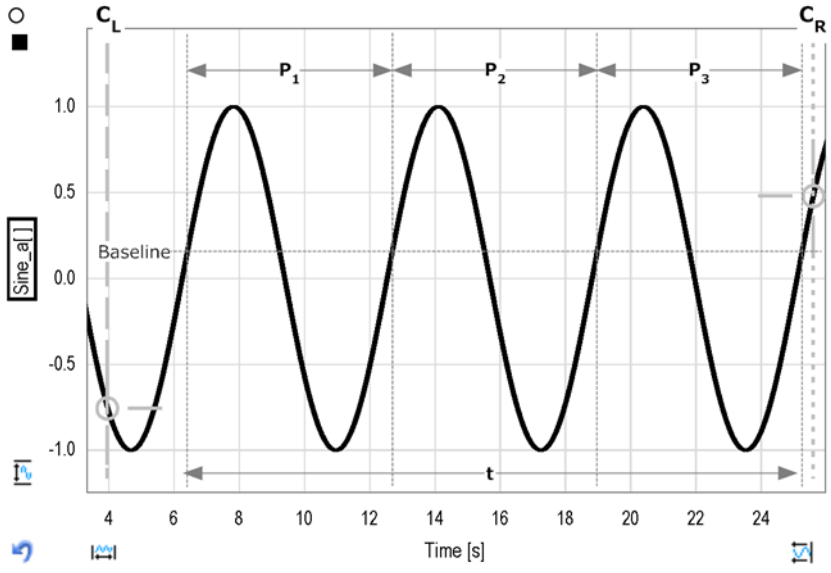


$$Duty\ Cycle = \frac{t}{P}$$

t Pulse Width at baseline level
P Period at baseline level

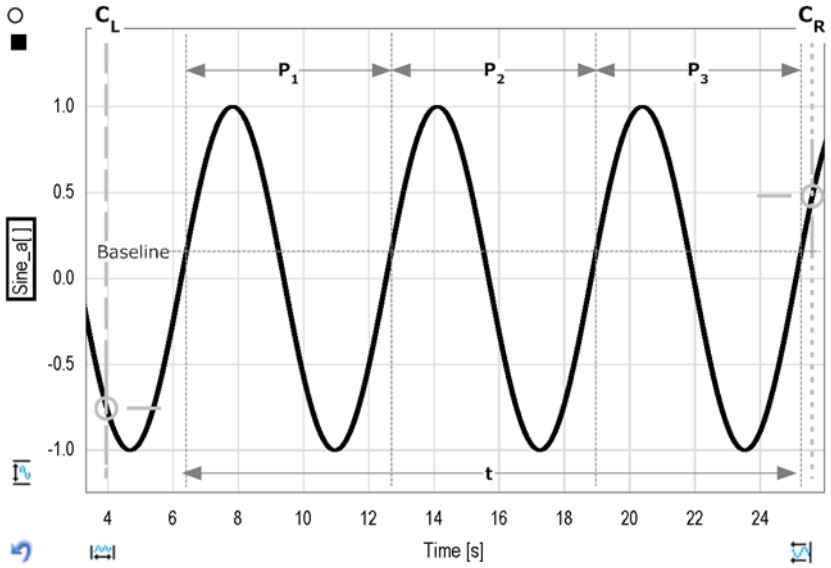
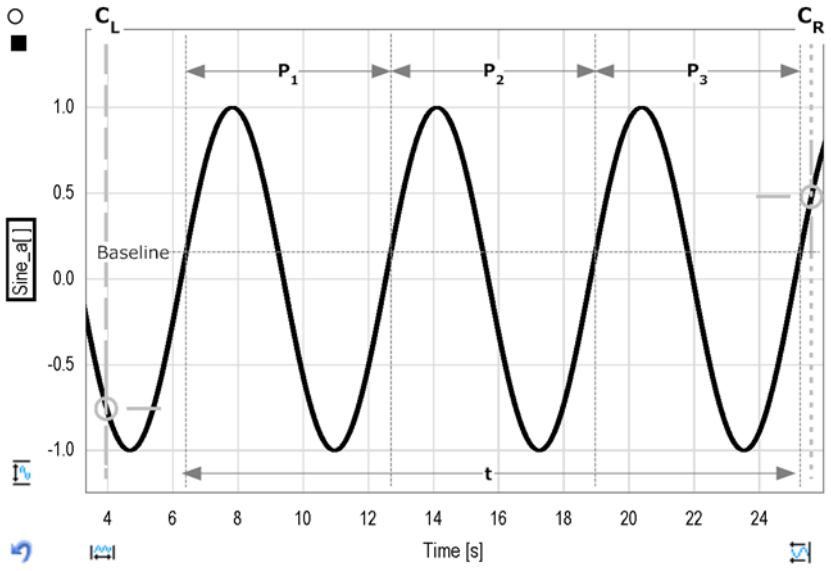
Frequency

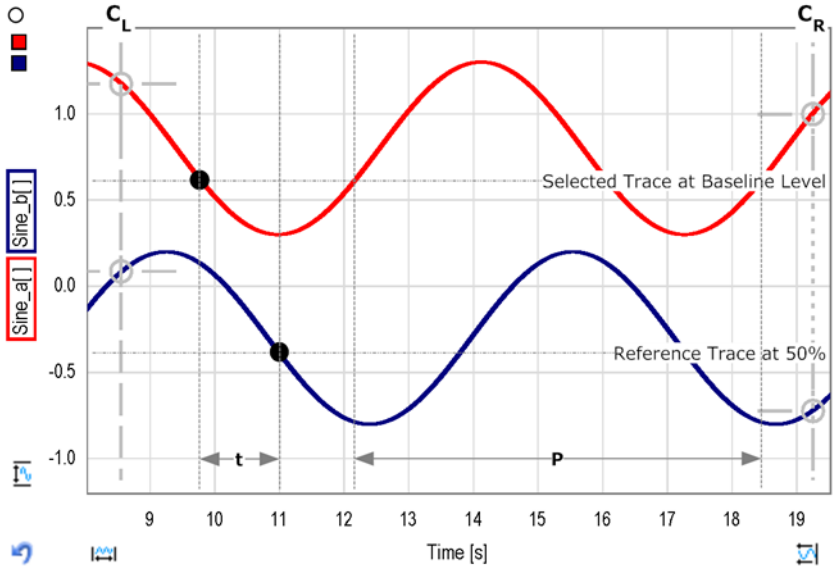
Sucht die Basislinie-Durchgänge des Signals und berechnet daraus die mittlere Frequenz. Die Durchgänge werden mit Hilfe der eingestellten Hysterese gegenüber Rauschen unempfindlicher gemacht.

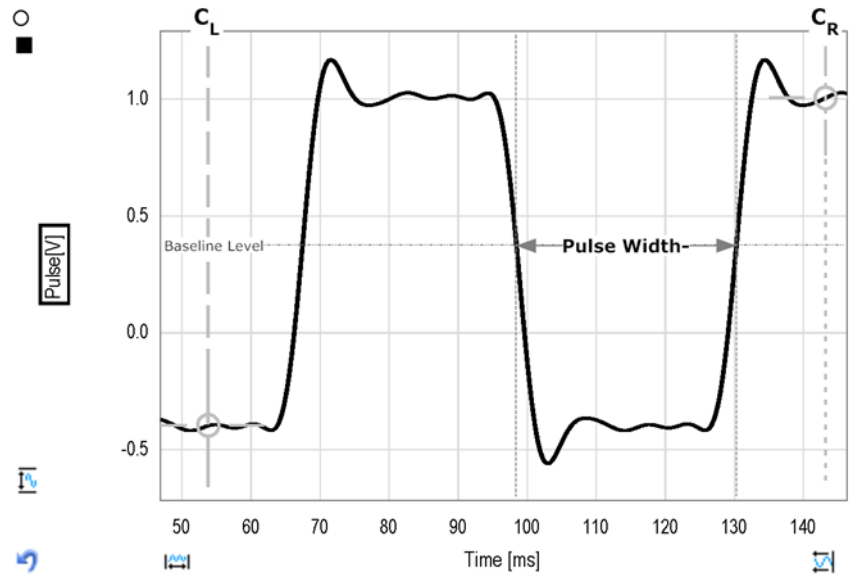


$$Frequency = \frac{N}{t}$$

C_L Cursor left
C_R Cursor right
P_x Period

	N Number of periods t Time between first and last baseline crossing
Number of Periods	Sucht die Basislinie-Durchgänge des Signals und zählt die Perioden. Die Durchgänge werden mit Hilfe der eingegebenen Hysterese gegen Rauschen unempfindlicher gemacht.  $Number\ of\ Periods = \frac{t}{P}$ C_L Cursor left C_R Cursor right P_x Period t Time between first and last level crossing
Period	Sucht die Basislinie-Durchgänge des Signals und berechnet daraus die mittlere Periodendauer. Die Durchgänge werden mit Hilfe der eingegebenen Hysterese gegen Rauschen unempfindlicher gemacht. 

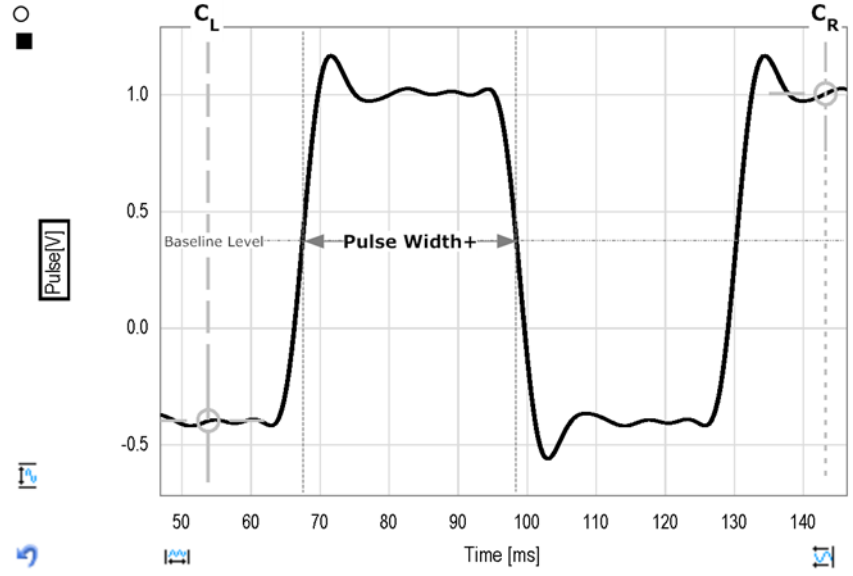
	$$Period = \frac{t}{N}$$ C_L Cursor left C_R Cursor right N Number of Periods t Time between first and last level crossing
Phase	Berechnet die Phasenlage von zwei periodischen, frequenzgleichen Signalen (normalerweise Spannung und Strom). Die Hauptkurve wird an der eingestellten Basislinie mit Hysterese ausgewertet. Die Referenzkuve kann vom entsprechenden Skalarmenü ausgewählt werden und wird bei 50% der Amplitude ausgewertet. Die Basislinien definieren die Pegel zur Bestimmung von ganzen Perioden. Für diese Funktion sollte **bevorzugt Skalartabelle B** verwendet werden. Die Berechnung erfordert **mindestens je 1 Periode**. Ein **negatives** Resultat wird zurückgegeben, wenn das **Referenz-Signal nacheilt**. Der Wertebereich der Phase ist ±180°.  $$Phase = 360 \cdot \frac{t}{P}$$ C_L Cursor left C_R Cursor right P Period t Time difference at baseline level to 50%
Pulse Width (neg)	Berechnet die negative Pulsbreite beim Pegel der Basislinie. Falls mehrere Perioden zwischen den Cursor liegen, wird die mittlere Pulsbreite berechnet.



C_L Cursor left
C_R Cursor right
Pulse Width- Negative Pulse Width at baseline level

Pulse Width (pos)

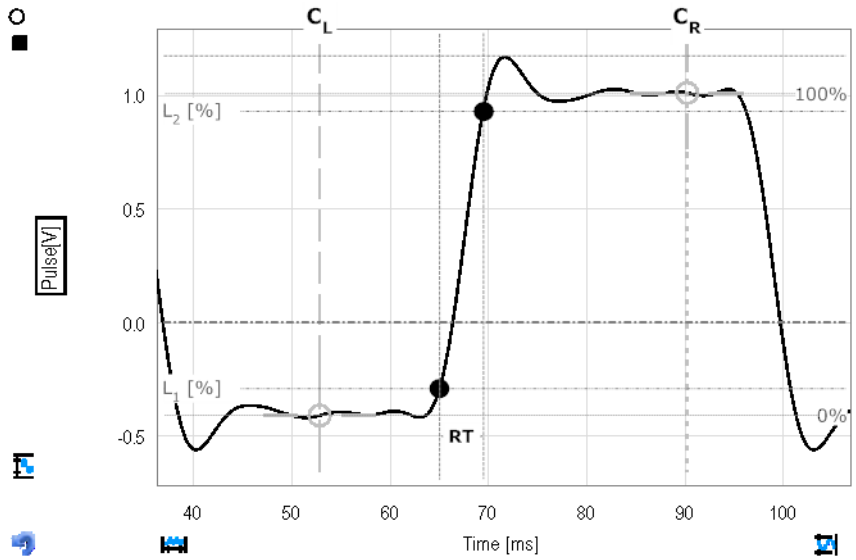
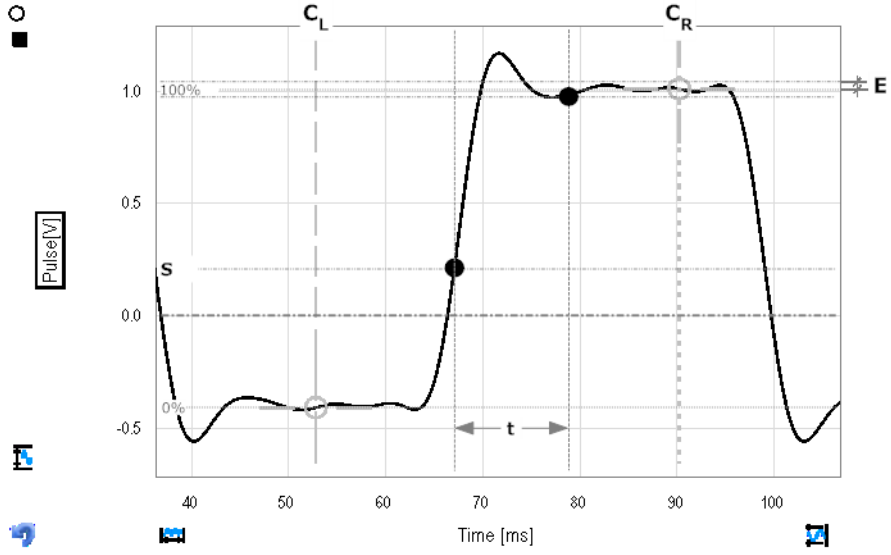
Berechnet die positive Pulsbreite beim Pegel der Basislinie. Falls mehrere Perioden zwischen den Cursor liegen, wird die mittlere Pulsbreite berechnet.



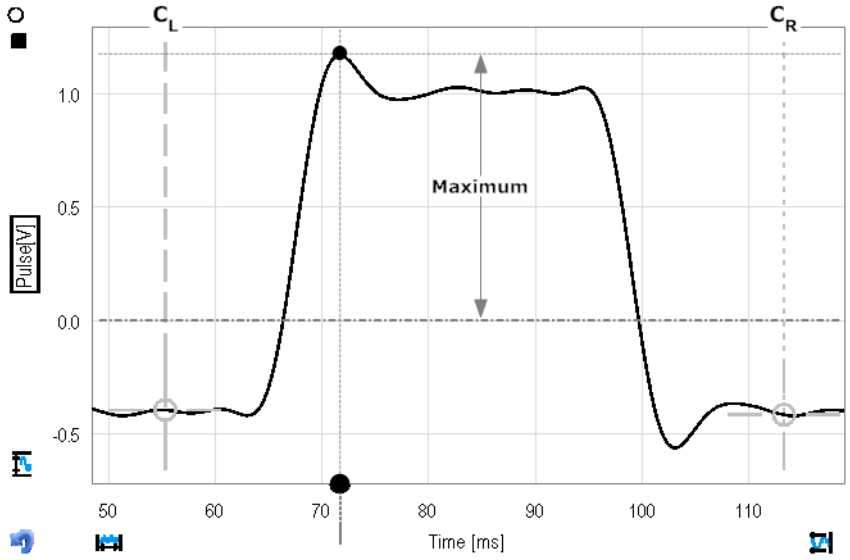
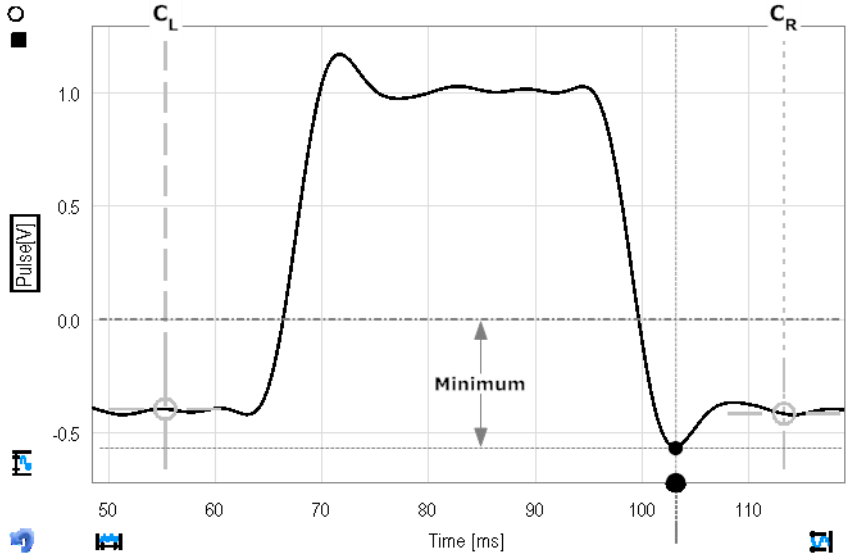
C_L Cursor left
C_R Cursor right
Pulse Width+ Positive Pulse Width at baseline level

Rise/Fall time

Ermittelt die Anstiegs- oder Abfallzeit einer Kurve bei gewünschten Pegeln [%].
Um die Anstiegszeit zu berechnen soll der linke Cursor auf die Grundlinie (0%) und der rechte Cursor auf die Schulter (100%) des Pulses positioniert werden.
Um die Abfallzeit zu berechnen soll der linke Cursor auf die Schulter (100%) und der rechte Cursor auf die Grundlinie (0%) des Pulses positioniert wer-

	den. Für diese Skalar-Funktion können (müssen) zwei Pegel eingestellt werden (z.B. 10% und 90%). Diese bestimmen die Punkte auf der Kurvenflanke zur Ermittlung der Anstiegs- bzw. Abfallzeit.  C_L Cursor left (0% amplitude) C_R Cursor right (100% amplitude) L_1 [%] Level 1 (Lower Level [%] for Rise Time) L_2 [%] Level 2 (Upper Level [%] for Rise Time) RT Rise time
Settling time	Gibt die Zeit zurück, ab welcher das Signal innerhalb eines Fehlerbandes bleibt. Das Fehlerband bezieht sich auf den Endwert (100%), der durch den rechten Cursor festgelegt wird.  C_L Cursor left (0%) C_R Cursor right (100%) S Starting point level [%] E Ending point level - Half error band [%] t Settling time

© Elsys AG

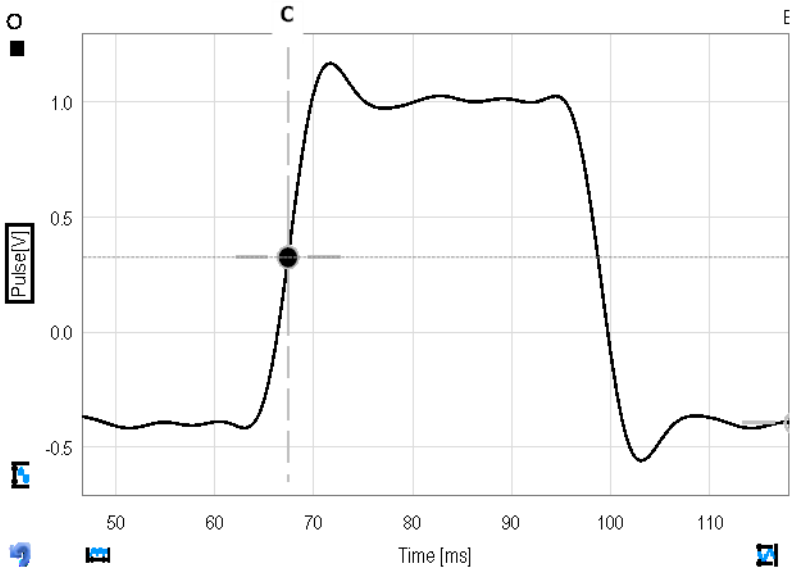
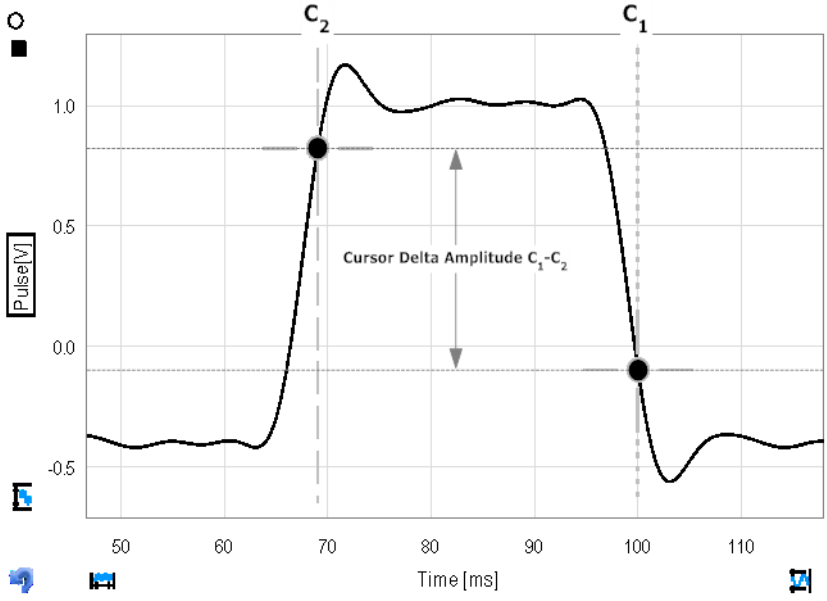
Time at Maximum	Sucht das Maximum und gibt die Zeit an der Position des Maximums zurück. Der Nullpunkt **hängt von den Einstellungen der X-Achse ab** wie *Relative Zeit* oder *Relative Zeit mit Nullpunkt beim Trigger*.  C_L Cursor left C_R Cursor right
Time at Minimum	Sucht das Minimum und gibt die Zeit an der Position des Minimums zurück. Der Nullpunkt **hängt von den Einstellungen der X-Achse ab** wie *Relative Zeit* oder *Relative Zeit mit Nullpunkt beim Trigger*.  C_L Cursor left C_R Cursor right

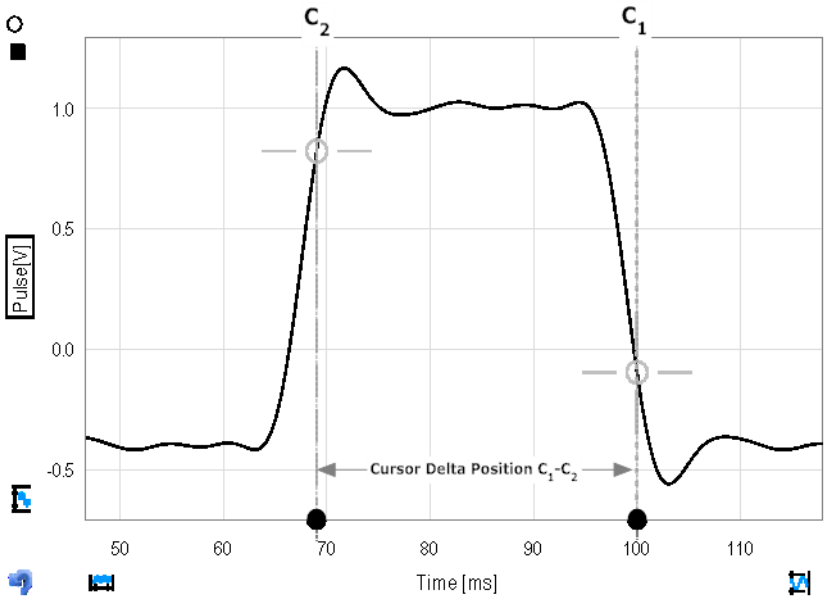
28.4 Gruppe "Periodic"

Crest Factor Periodic	Diese Funktion ist in der Gruppe "Power" beschrieben.
Mean Periodic	Diese Funktion ist in der Gruppe "Vertical" beschrieben.
Rectified Mean periodic	Diese Funktion ist in der Gruppe "Vertical" beschrieben.
RMS Periodic	Diese Funktion ist in der Gruppe "Vertical" beschrieben.
Standard Deviation Periodic	Diese Funktion ist in der Gruppe "Vertical" beschrieben.

© Elsys AG

28.5 Gruppe "Cursor"

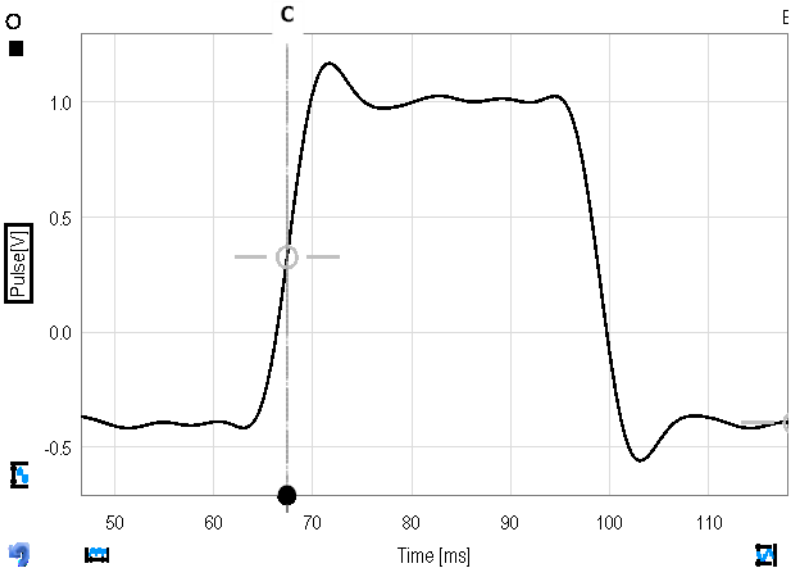
Cursor Amplitude	Gibt den Amplituden-Wert der Kurve an der Position des selektierten Cursors aus.  $Cursor\ Amplitude = y_{Cursor}$
Cursor Delta Amplitude	Gibt die Amplituden-Differenz der beiden selektierten Cursor aus.  $Cursor\ Delta\ Amplitude = y_{Cursor\ 1} - y_{Cursor\ 2}$
Cursor Delta Position	Gibt die Differenz der beiden Cursor-Positionen zurück.



$$Cursor\ Delta\ Position = x_{Cursor\ 1} - x_{Cursor\ 2}$$

Cursor Position

Gibt die Position des selektierten Cursors auf der X-Achse zurück.
Der Nullpunkt hängt von den Einstellungen der X-Achse ab wie Relative Zeit oder Relative Zeit mit Nullpunkt beim Trigger.



$$Cursor\ Position = x_{Cursor}$$

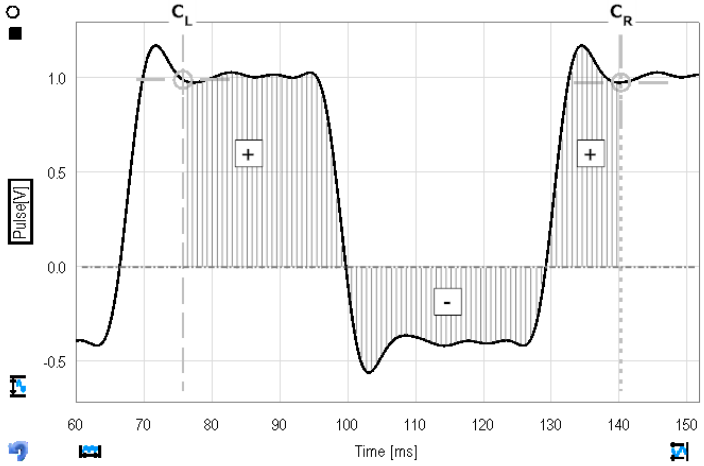
28.6 Gruppe "Power"

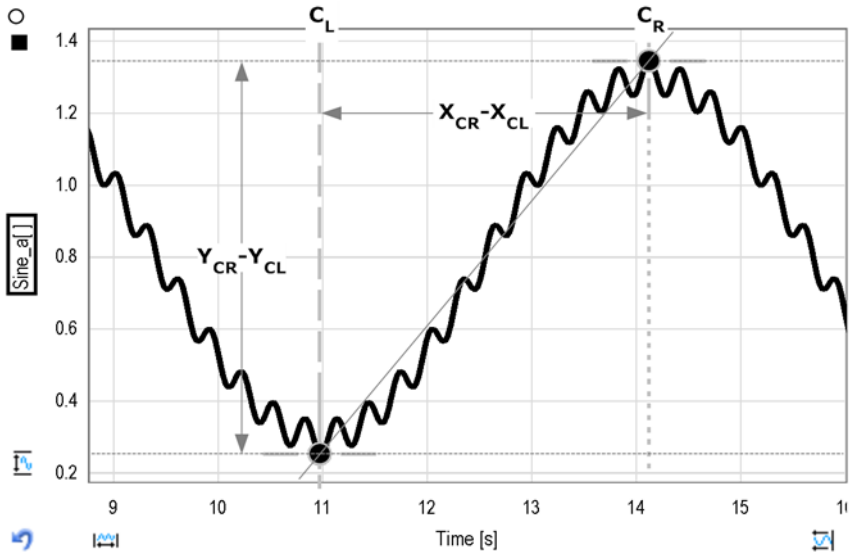
Apparent Power	Berechnet die Scheinleistung von zwei periodischen Signalen aus Spannung und Strom. $Apparent\ Power = RMS_{Periodic}\ (i) \cdot RMS_{Periodic}\ (u)$ $RMS_{Periodic}$ See Scalar RMS Periodic i Current waveform u Voltage waveform
Cos (phi)	Berechnet den Cosinus-Phi aus den Grundwellen von zwei periodischen, frequenzgleichen Signalen (normalerweise Spannung und Strom). Die Hauptkurve wird an der eingestellten Basislinie mit Hysterese ausgewertet. Die Referenzkurve kann im entsprechenden Skalarmenü ausgewählt werden und wird bei 50% der Amplitude ausgewertet. Die Basislinien definieren die Pegel zur Bestimmung von ganzen Perioden. Für diese Funktion sollte **bevorzugt Skalartabelle B** verwendet werden. Die Berechnung erfordert mind. **je 1 ganze Periode**. Beim Einsatz von Spannungssonden und Stromzangen können Zeitverschiebungen auftreten verursacht durch verschiedene Bandbreiten und Kabellängen. Solche Phasenverschiebungen sollten kompensiert werden bevor cos (phi) berechnet wird. Die Formel-Funktion Shift kann beigezogen werden, um eine Kurve gegen die andere zeitlich zu verschieben. $\cos(\varphi) = \cos\ (360 \cdot \frac{T}{P})$ P Period T Time difference of the rising edges
Crest Factor	Der Crest-Faktor beschreibt das Verhältnis zwischen Spitzenwert und RMS. $Crest\ Factor = \frac{\lvert Peak \rvert}{RMS}$ Peak Absolute Peak RMS Root Mean Square
Crest Factor Periodic	Der Crest-Faktor beschreibt das Verhältnis zwischen Spitzenwert und RMS von periodischen Signalen. $Crest\ Factor_{Periodic} = \frac{\lvert Peak \rvert}{RMS_{Periodic}}$ Peak Absolute Peak $RMS_{Periodic}$ Periodic Root Mean Square
Fundamental Power	Sucht die Basislinie-Durchgänge der Spannungs- und der Stromkurve, um die Periodendauer zu bestimmen und berechnet anschliessend daraus Fun-

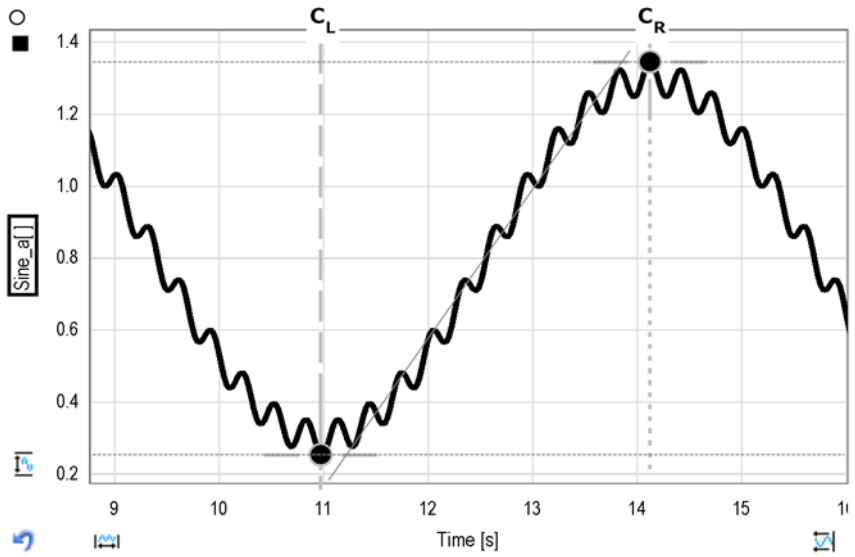
	damental Power, die Wirkleistung in der Grundwelle ohne die Oberwellen. Es wird davon ausgegangen, dass die beiden Signale periodisch sind und dass es sich dabei um ein Spannungs- und ein Stromsignal handelt.
Phase	Diese Funktion ist in der Gruppe "Horizontal" beschrieben.
Power Factor	Berechnet den Leistungs-Faktor aus zwei periodischen Signalen mit gleicher Frequenz (normalerweise Spannung und Strom). Das Referenz-Signal kann aus der Dropdown-Liste ausgewählt werden. Für diese Funktion sollte **bevorzugt *Skalartabelle B*** verwendet werden. Die Berechnung erfordert mind. 1.5 ganze Perioden. Zwischen Spannungs- und Stromkurve können Zeitverschiebungen auftreten, die von unterschiedlichen Bandbreiten und Kabellängen der Tastköpfe herrühren. Ein solcher Zeitversatz entspricht einem Phasenfehler und muss korrigiert werden bevor diese Berechnung durchgeführt wird. Der Formeleditor bietet die Funktion *Shift* an, die zur zeitlichen Verschiebung einer Kurve beigezogen werden kann. $Power\ Factor = \frac{P}{S}$ P Real Power S Apparent Power
Reactive Power	Berechnet die Blindleistung aus Scheinleistung und Wirkleistung. Für diese Funktion sollte **bevorzugt *Skalartabelle B*** verwendet werden. $Preact = \sqrt{Papp^2 - Preal^2}$ Preact Reactive Power Papp Apparent Power Preal Real Power
Real Power	Berechnet die Wirkleistung aus den einzelnen Momentanleistungs-Samplings. Für diese Funktion sollte **bevorzugt *Skalartabelle B*** verwendet werden. $Preal = Mean_{Periodic}(i \cdot u)$ Preal Real Power $Mean_{Periodic}$ See Scalar Mean Periodic i Current waveform u Voltage waveform
RMS	Diese Funktion ist in der Gruppe "Vertical" beschrieben.
RMS Periodic	Diese Funktion ist in der Gruppe "Vertical" beschrieben.
Total Harmonic Distortion	Sucht zuerst die Grundfrequenz und berechnet dann den Wert THD, Total Harmonic Distortion (THD) in % eines periodischen Signales. Zur Berechnung werden mindestens 2 Perioden der dominanten Frequenz zwischen dem Cursor-Paar benötigt. Verrauschte Signale müssen eventuell vor der Berechnung mittels eines Tiefpass-Filters gefiltert werden. Auch sollten die **Basislinie und vor allem**

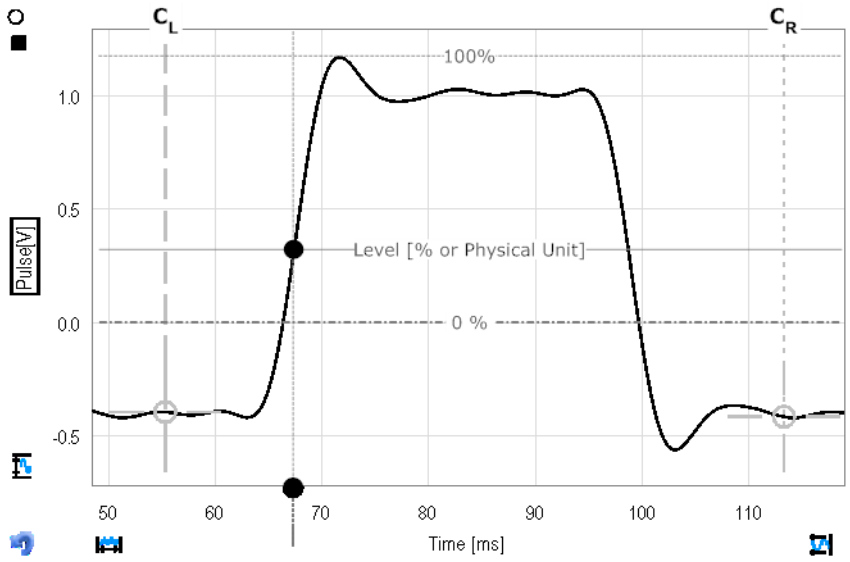
	die Hysterese sorgfältig gewählt werden. $$THD_U = \frac{\sqrt{U_2^2 + U_3^2 + U_4^2 + \cdots U_n^2}}{U_1} [\%]$$ $$THD_I = \frac{\sqrt{I_2^2 + I_3^2 + I_4^2 + \cdots I_n^2}}{I_1} [\%]$$ U_1 Voltage at fundamental frequency I_1 Current at fundamental frequency U_x x_{th} harmonic of the voltage signal I_x x_{th} harmonic of the current signal

28.7 Gruppe "Misc"

Area	Berechnet die Fläche unter der Kurve zwischen den Cursor relativ zum Nullpunkt. Werte oberhalb der Nulllinie werden als positive Flächenwerte berechnet, Werte unterhalb der Nulllinie als negative Flächenwerte.  $Area = \sum_{i=C_L}^{C_R} y_i \cdot \Delta t$ C_L Cursor left C_R Cursor right y_i y-value at position i Δt Sampling Interval
Cadence	Berechnet die Frequenz normiert auf die Minute. $Cadence = Frequency \cdot 60$ Frequency Calculated Scalar
Energy	Berechnet das Integral der quadrierten Punkte. $Energy = \sum_{i=C_L}^{C_R} y_i^2 \cdot \Delta t$ C_L Cursor left C_R Cursor right y_i y-value at position i Δt Sampling Interval
Number of Blocks	Gibt die totale Anzahl Blöcke an, auch während die Erfassung läuft. Dieser Wert ist Cursor-unabhängig. Number of Blocks wird vor allem im Multiblock- und ECR-Datenerfassungs-Mode gebraucht.
Number of Triggers	Gibt die totale Anzahl Trigger-Ereignisse an.

	Dieser Wert ist Cursor-unabhängig. Number of Triggers wird vor allem im Multiblock- und ECR-Datenerfassungs-Mode gebraucht.
OverFlow	OverFlow beobachtet den positiven Überlauf des ADCs des Eingangsverstärkers. Gibt *"No"* in der Hintergrundfarbe *Grün* zurück falls kein Überlauf festgestellt wird. Ansonsten wird *"Yes"* mit der Hintergrundfarbe *Rot* und der Angabe des Zeitpunktes des ersten Überlaufes angezeigt. Siehe auch UnderFlow.
Slope	Berechnet die Steigung zwischen den Schnittpunkten der Kurve mit den Cursors.  $Slope = \frac{y_{CR} - y_{CL}}{x_{CR} - x_{CL}}$ y_{CL} y-value at position cursor left y_{CR} y-value at position cursor right x_{CL} x-value at position cursor left x_{CR} x-value at position cursor right
Slope (Lin. Regr.)	Berechnet die Steigung der Kurve mittels linearer Regression zwischen den Cursor.

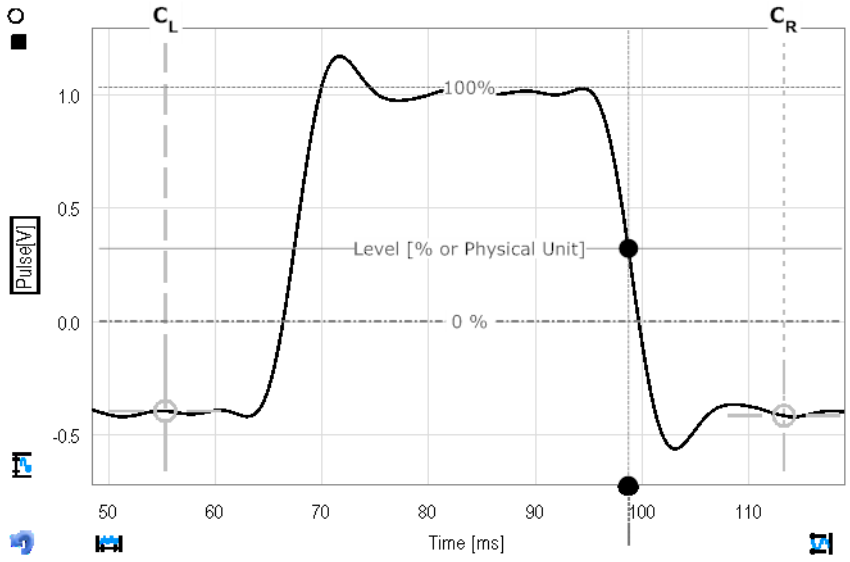
	 *Slope by Linear Regression* C_L Cursor left C_R Cursor right
Text	Das ist ein simpler Platzhalter. Er ermöglicht einen beliebigen Text in der Tabelle einzutragen; zwecks Reports oder zur Strukturierung der Tabelle. **Diese Funktion ist nur bei Skalarfunktionen Tabelle B vorhanden!**
Trigger Real Time	Gibt die Zeit des Trigger-Ereignisses in relativer oder absoluter Zeit zurück. In den Betriebsarten Multi-Block oder ECR wird das Resultat aus dem nächstgelegenen Datenblock zwischen den angewählten Cursor ermittelt. Im Scope- und kontinuierlichen Mode liefern nur Resultate in absoluter Zeit sinnvolle Ergebnisse.
TxLeft	Gibt die Zeit zurück, bei der in einem positiven Puls ein zu definierender Pegel ansteigend überschritten wird. In diesem Fall wird vom linken Cursor ausgegangen. Der Pegel kann in % vom Maximum (Null-Linie entspricht 0%, Maximum entspricht 100%) oder als physikalische Grösse angegeben werden. Falls *Abs. Max* angewählt ist, so wird zuerst die absolute Kurve berechnet bevor TxLeft ermittelt wird. Der Nullpunkt hängt von den Einstellungen der X-Achse ab wie *Relative Zeit* oder *Relative Zeit mit Nullpunkt beim Trigger*.



C_L Cursor left
C_R Cursor right

TxRight

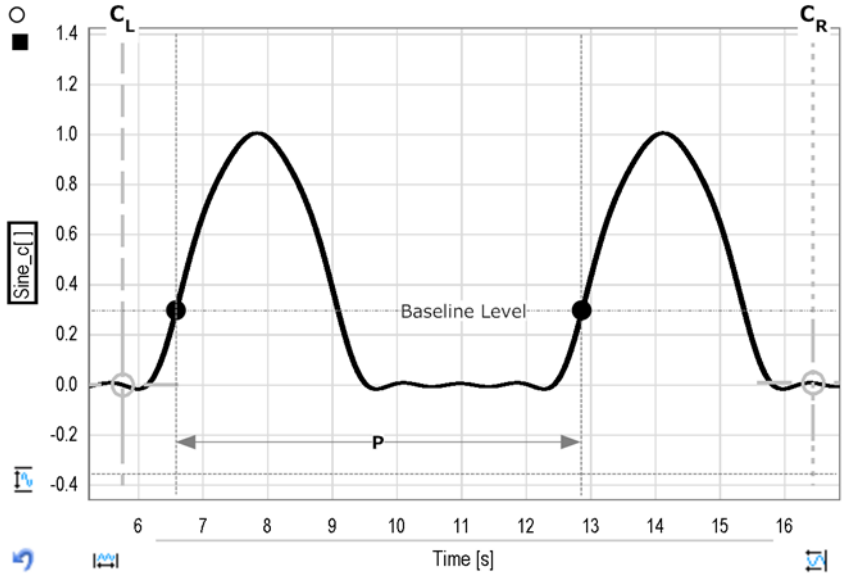
Gibt die Zeit zurück, bei der in einem positiven Puls ein zu definierender Pegel ansteigend überschritten wird. Wobei in diesem Fall vom rechten Cursor ausgehend rückwärts gesucht wird. Der Pegel kann in % vom Maximum (Null-Linie entspricht 0%, Maximum entspricht 100%) oder als physikalische Grösse angegeben werden.
Falls *Abs. Max* angewählt ist, so wird zuerst die absolute Kurve berechnet bevor TxLeft ermittelt wird.
Der Nullpunkt hängt von den Einstellungen der X-Achse ab wie *Relative Zeit* oder *Relative Zeit mit Nullpunkt beim Trigger*.



C_L Cursor left
C_R Cursor right

UnderFlow

UnderFlow beobachtet den negativen Überlauf des ADCs des Eingangsverstärkers.
Gibt *"No"* in der Hintergrundfarbe *Grün* zurück falls kein Überlauf festgestellt wird. Ansonsten wird *"Yes"* mit der Hintergrundfarbe *Rot* und der An-

	gabe des Zeitpunktes des ersten Überlaufes angezeigt. Siehe auch Over-Flow.
Velocity	Berechnet die Geschwindigkeit aus Periodendauer und vorgegebener Distanz.  $Velocity = \frac{Sensor\ Distance}{P}$ Sensor Distance Sensor movement [m] P Period

29 Diverses

Weitere Informationen und Beschreibung von Funktionen und Formeln

29.1 ActiveX/COM - Interface

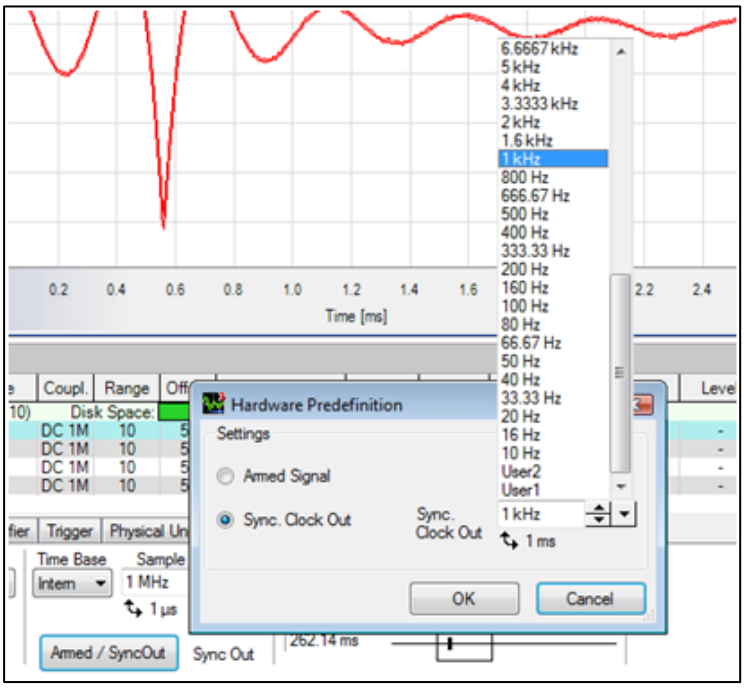
Die Active-X/COM - Schnittstelle ist als Option zu TranAX erhältlich.

Die COM (Component Object Model), aus von Microsoft, ist eine Möglichkeit, wie Komponenten (Programme) untereinander kommunizieren können. Diese Möglichkeit wird verwendet, um über Drittprogramme (Bsp. BallAX oder Excel Visual-Basic) TranAX zu steuern.

Die genaue Verwendung und Einstellungen der Parameter können dem ActiveX-Handbuch entnommen werden.

29.2 Sync.Clock Out

Ab Firmware Version 4 des Sternverteilers steht die Möglichkeit des **Sync.Clock Out** zur Verfügung. Diese Funktion kann dazu verwendet werden, **externe Geräte** wie z.B. eine Highspeed Kamera zu **synchronisieren**.



Die Frequenz des Sync.Clock-Ausgangs kann aus einer **Dropdown-liste** ausgewählt oder **manuell** eingegeben werden. Der eingegeben Wert wird dann auf den nächsten passenden gerundet.
Es stehen Frequenzen zwischen **10Hz und 10MHz** zur Verfügung. Die beiden Einträge in der Liste, "User 1" und "User 2" sind reserviert und können vor der Auslieferung) auf **Frequenzen gemäss Kundenwunsch programmiert** werden.
Beim Karteireiter ***"Haupt-Einstellungen"*** im **Control Panel** kann durch klicken des Knopfs ***"Armed/SyncOut"*** der Dialog ***"Hardware Predefinition"*** geöffnet und zwischen **Armed Signal** und **Sync. Clock Out** gewählt werden. Normalerweise wird das Sync.Clock- bzw. das Armed-Signal am 25pol. D-Sub-Stecker ausgegeben (siehe Hardware-Beschreibung).

Um die **Sync.Clock Out** Funktion zu verwenden, muss die Installierte Software eventuell aktualisiert werden. Die folgenden Versionen werden als **Minimum vorausgesetzt**:

TranAX:	**3.2.1.702**	(Menü ***"Hilfe" / "Info"***)
TPC-Server:	**1.3.5**	(Control Panel /)
Sternverteiler:	**4**	(TraNetConfiguration.exe / Show Logfile / Server)

Ein Update der Firmware des Sternverteilers muss im Lieferwerk erfolgen!

Bei hohen Frequenzen (über 1MHz**)** muss auf die **Länge des Kabels** geachtet werden (< 2m bei 10MHz). Ansonsten sind die Signale beim Endgerät nicht mehr zuverlässig verwendbar.

Die Frequenzen für "User 1" und "User 2" müssen mit einer **angepassten Firmware kundenspezifisch einprogrammiert** werden und können nicht ad hoc geändert werden.
Für Änderungen (wie auch bei Updates von Geräten die vor 2012 ausgeliefert wurden) muss der Sternverteiler zurück ans Lieferwerk gesendet werden.

29.3 Konsolen Parameter

TranAX lässt sich auch aus der Konsole oder einer Batch-Datei starten. Die vorhandenen Parameter bieten dem routinierten Anwender eine Fülle von Optionen und Möglichkeiten.

Diese Beschreibung ist für **fortgeschrittene Anwender** und setzt ein bestimmtes **Grundwissen** im Umgang mit der Konsole und Batch-Dateien voraus. Es wird nicht weiter in die Details der jeweiligen Windows Versionen eingegangen. **Bei Fragen wenden Sie sich bitte an Ihre interne IT-Abteilung.**

Um die möglichen Parameter in der Konsole aufzulisten, wechseln Sie in das TranAX-Verzeichnis (in der Regel "C:\Program Files\Elsys\TranAX3") und geben Sie ***TranAX.exe -help*** ein.

Die untenstehende Liste Zeigt die die Ausgabe und verfügbaren Parameter diese werden unabhängig der eingestellten Sprache immer in Englisch ausgegeben:

Parameter	Beschreibung und Beispiel
-allsettings	Loads all settings from the specific path. If the file name or directory path has blanks you have to write the whole path with quotes EXAMPLE: TranAX.exe -allsettings="C:\YOURPATH\MyAllSettings.lay"
-autoseq	Loads the autosequence from the specific path and starts it. If the file name or directory path has blanks you have to write the whole path with quotes EXAMPLE: TranAX.exe -autoseq="C:\YOURPATH\MyAutoSeq.aut"
-cachesize	Limits the data cache to the specified amount of bytes (default 150'000'000) EXAMPLE: TranAX.exe -cachesize=200000000
-device	Sets url of device(s) to use EXAMPLE: TranAX.exe -device=192.168.0.102:10010
-experiment	Opens the given experiment. If the file name or directory path has blanks you have to write the whole path with quotes EXAMPLE: TranAX.exe -experiment="C:\YOURPATH\Experiment.exp"
-formula	Loads the formula file from the specific path. If the file name or directory path has blanks you have to write the whole path with quotes EXAMPLE: TranAX.exe -formula="C:\YOURPATH\MyFormula.for"
-help	Shows all commands EXAMPLE: TranAX.exe -help
-info	Shows metadata of a TPC5-File. If the file name or directory path has blanks you have to write the whole path with quotes EXAMPLE: TranAX.exe -info=C:\YOURPATH\test.tpc5
-layout	Loads the layout from a specific path. If the file name or directory path has blanks you have to write the whole path with quotes EXAMPLE:

	TranAX.exe -layout="C:\YOURPATH\MyLayout.lay"
-recording	Loads the recording settings from the specific path. If the file name or directory path has blanks you have to write the whole path with quotes EXAMPLE: TranAX.exe -recording="C:\YOURPATH\MyRecording.tps.xml"
-server	Acts as remote server (on port 12668) EXAMPLE: TranAX.exe -server
-serverport	Acts as remote server with a defined port EXAMPLE: TranAX.exe -serverport=12000
-title	Sets the title of the TranAX program. This could be necessary if you open several TranAX program EXAMPLE: TranAX.exe -title=VoltageMeasurement
-version	Shows the current build version of TranAX EXAMPLE: TranAX.exe -version
-view	Adds a file to the signal source browser. If the file name or directory path has blanks you have to write the whole path with quotes EXAMPLE: TranAX.exe -view="C:\YOURPATH\test.tpc5"
-offline	Starts TranAX without connecting to a device EXAMPLE: TranAX.exe –offline
-viewer	Starts TranAX as a viewer. The user can not manipulate measurement configurations (Start, stop, etc.) EXAMPLE: TranAX.exe -viewer
-scope	Starts TranAX in fullscreen mode with a Scope-Display EXAMPLE: TranAX.exe -scope
-fullscreen	Starts TranAX in fullscreen mode EXAMPLE: TranAX.exe -fullscreen

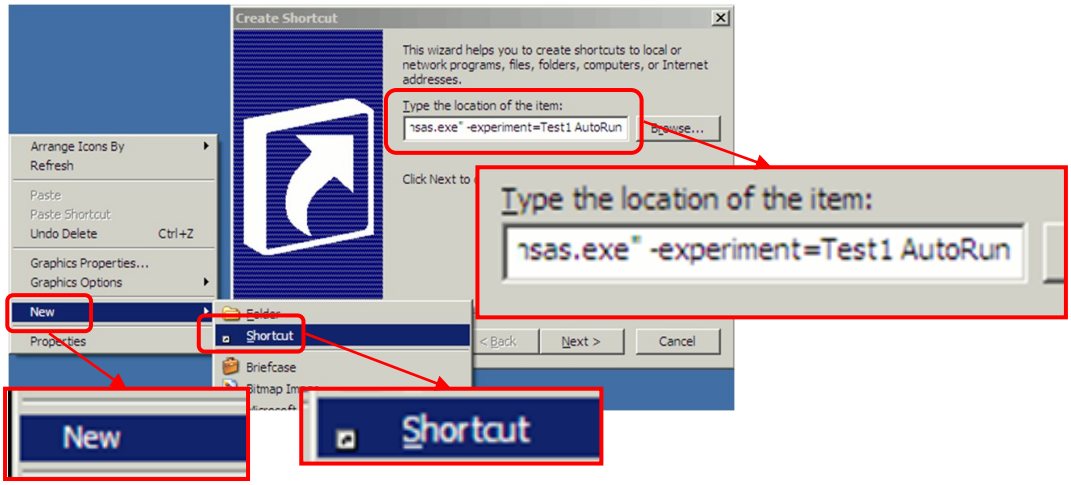
29.4 Verknüpfungen erstellen

Sie können TranAX-Experimente auch mit **(Desktop)-Verknüpfungen starten**. Um TranAX direkt mit einem bestehenden Experiment zu laden, fügen Sie der Kommandozeile das folgende Kommando hinzu: ***-experiment="experiment name"*** Falls noch zusätzlich eine Autosequenz gestartet werden soll, geben Sie den entsprechenden Namen an:
-autosequence="autosequence name"

Beispiel:

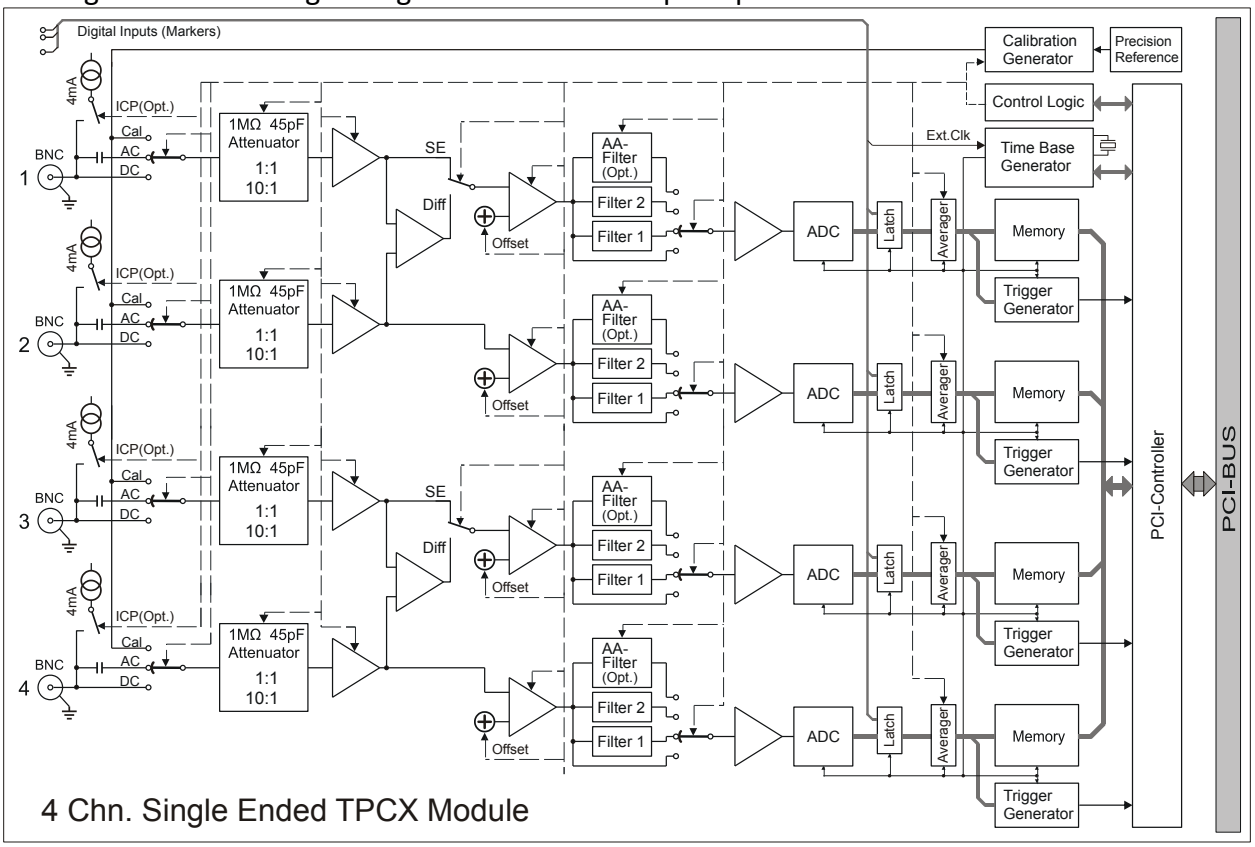
```
"C:\Program Files\Elsys\TRANAX\TranAX.exe" -experiment=Test1.EXP
-autosequence=AutoRun.AUT
```

Sie können verschiedene Verknüpfungen für Ihre jeweilige Aufgabe erstellen. Um eine Desktop-Verknüpfung für TranAX zu erstellen, klicken Sie mit der **rechten Maustaste** auf dem **Windows-Desktop** und wählen Sie im Kontextmenü "*Neu*" / "*Verknüpfung*". Navigieren Sie zu C:\Program Files\Elsys\TRANAX\ und wählen Sie *TranAX.exe*. Um nun ein Experiment oder eine Autosequenz automatisch zu laden, fügen Sie wie im obigen Beispiel die Parameter an.

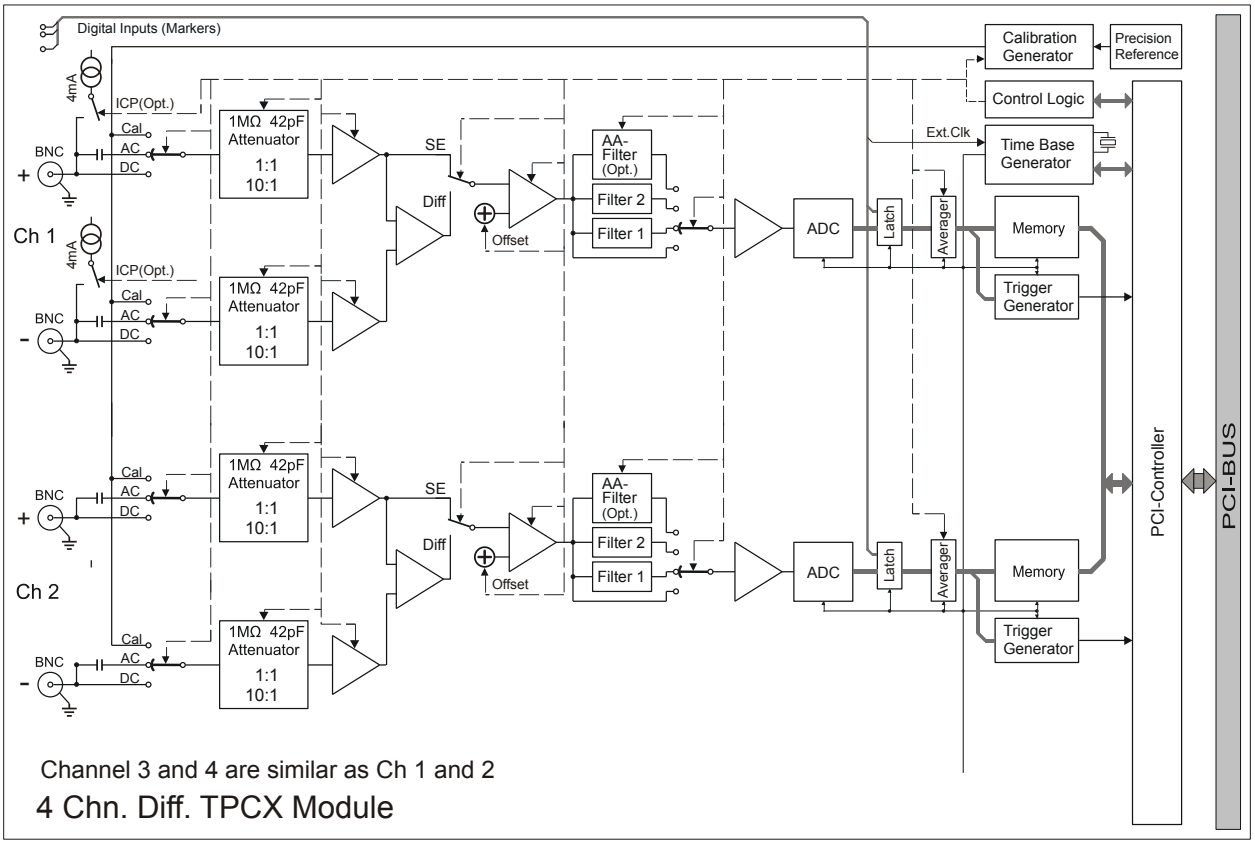


29.5 Blockschaltbilder

Die folgenden Abbildungen zeigen vereinfacht den prinzipiellen Aufbau eines TPCX-Modules.



4 Chn. Single Ended TPCX Module



Channel 3 and 4 are similar as Ch 1 and 2

4 Chn. Diff. TPCX Module

Die Signale gelangen über **massebezogene BNC-Eingänge** in die TPCX/TPCE-Karte. Zuerst werden die einstellbaren Eingangsverstärker dem jeweiligen **Eingangsspannungsbereich** angepasst. Danach können individuell via Software je zwei **benachbarte Eingänge** zu einer **differentiellen Eingangsstufe** zusammengeschaltet werden.

Um den **Nullpunkt** einzustellen, wird in der 2. Stufe sofwaregesteuert eine **Offsetspannung** zum Signal addiert. Zwischen der 2. Stufe und den ADC-Vorverstärkern können **Rauschfilter** oder optional Anti-Aliasing-Filter zugeschaltet werden. Der ADC digitalisiert die Analogsignale und sendet sie zum Speicher.

Die digitalen Werte werden einerseits auf die eingestellten **Triggerbedingungen** hin überprüft und andererseits in die karteneigenen Speicher geschrieben, von wo sie der Computer auslesen kann. Dieser ganze Ablauf wird vom **Timebase-Block** gesteuert.

Die **grosse Speichertiefe** (optional bis 128MSamples/Kanal) kann in kleinere Blöcke unterteilt werden. Damit lässt sich die Erzeugung von **unnötig grossen Datenmengen** vermeiden. Ausserdem wird damit die Aufzeichnung von mehreren schnell hintereinander auftretenden Ereignissen ermöglicht (**Multiblock**-Aufnahmen).

Jeder Kanal verwendet einen 14-Bit ADC und einen 16-Bit Wort Speicher. Die verbleibenden **2 Bits können als Marker** (digitale Signale) verwendet werden. Bei einem 16-Bit-Modul können die Marker auch aufgezeichnet werden, wenn im Control-Panel das Averaging auf 14 Bit gestellt wird.

29.6 Einschränkungen

Die folgenden Punkte behandeln gewisse Einschränkungen, die bei den Einstellungen beachtet werden müssen.

29.6.1 Digitale Eingänge (Marker)

Digitale Eingänge sind nur dann verfügbar, wenn optional die entsprechende **Marker-Hardware** installiert ist. Ein Marker-Signal steht immer in Bezug zum Namen eines Analog-Kanals. Jeder mit der Option ausgestattete Analog-Kanal enthält 2 Marker-Spuren. Die Software kann keine direkten Hinweise liefern bei welchen Kanälen Marker installiert sind. Bei 16-Bit Modulen können Marker nur aufgezeichnet werden, wenn der ADC auf 14-Bit gestellt wird (siehe Mittelung). Marker-Kanäle können nicht als Trigger verwendet werden, aber es kann ein externer digitaler Trigger-Eingang pro Gerät verwendet werden. Der externe Trigger wird via Trigger-Einstellungen gesteuert.

29.6.2 Differential-Eingänge

Mit den Single-Ended Modulen (z.B. TPCX-2014-**8S**) werden die folgenden Kanal-Paarungen für die **differentiellen Eingänge** verwendet: (1,2), (3,4), (5,6), (7,8). In diesem Fall sind die **geradzahligen Kanäle** nicht einstellbar.
Mit den differentiellen Modulen (z.B. TPCX-2014-**8D**) sind alle Kanäle mit zwei BNC-Steckern versehen, deshalb haben auch geradzahlige Kanäle differentielle Eingänge.
Die entsprechenden Einstellungen werden im Abschnitt Eingangs-Verstärker beschrieben.

29.6.3 Maximale Abtastrate

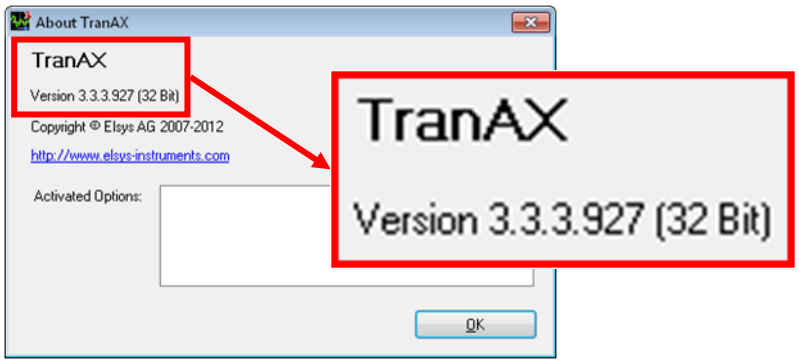
Die **vertikale Auflösung von 16-Bit** (mit den 16-Bit-Modulen wie TPCX-4016-4D) ist nur gegeben bei **Abtastraten gleich oder unterhalb 1/4 der maximalen Abtastrate**. Beispiel: Bei einer TPCX-4016-4x Modul (40MHz), ist die 16-Bit-Auflösung nur für Abtastraten gleich oder kleiner 10 MS/s gegeben. Oberhalb von 10 MS/s bis 40 MS/s entspricht die vertikale Auflösung dem 14-Bit ADC.

30 Fehlerbehebung

Um Fehlverhalten schnell zu lösen ist es entscheidend, dass der Support mit den wichtigsten Informationen versorgt wird. Dieses Kapitel beschreibt wie Probleme selbst gelöst werden können und welche Daten für allfällige Anfragen mitgegeben werden sollten.

30.1 TranAX Software Version

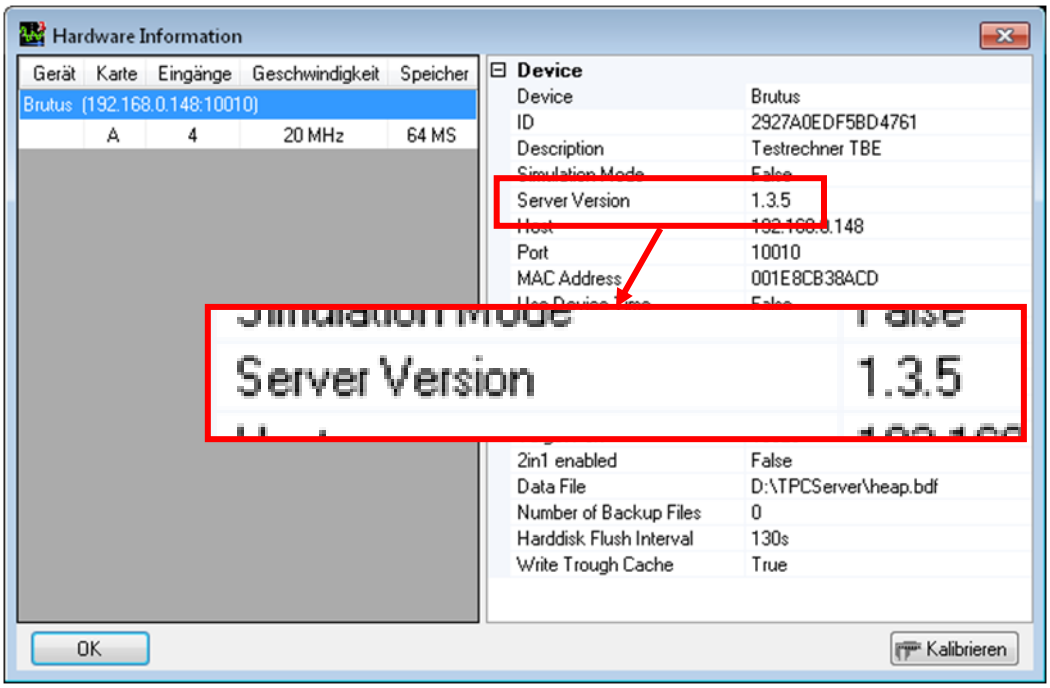
Um eine Messung in TranAX zu erfassen sind mehrere Software Komponenten miteinander in Verbindung. TranAX kommuniziert mit dem TPC-Server, dieser mit dem Treiber und dieser wieder mit den TPCX/TPCE Modulen.



Um die Version Ihrer TranAX Version zu bestimmen, klicken Sie im Menü ***"Hilfe" / "Info"***. In diesem Beispiel handelt es sich um die Version 3.3.3.927.

30.2 TPC-Server Version

Die Version des TPC-Servers kann wie folgt bestimmt werden: Klicken Sie im Control Panel auf den Information Knopf , anschliessend wir das folgenden Fenster geöffnet. Hier ist die Server Version 1.3.5 installiert.

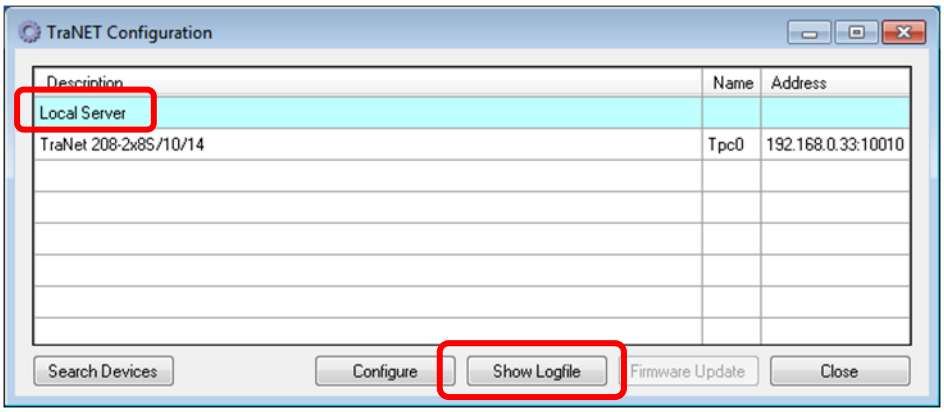


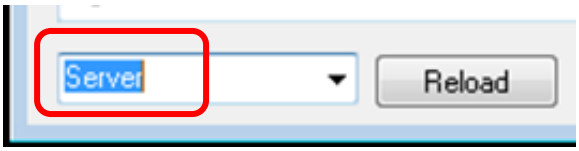
30.3 Treiber und Firmware Version

Um die Version der Treiber, der Firmware für die TPCX/TPCE -Module und des Sternverteilers zu ermitteln, muss erst TraNetConfiguration.exe gestartet werden. Dazu doppel klicken Sie das folgende Symbol auf dem Desktop.

Anschliessend wählen Sie im "TraNET Configuration" Dialog Ihr Gerät, entweder "Local Server" oder eines der gefunden TraNET FE Geräte, aus. Anschliessend klicken Sie auf den Show Logfile Knopf.

Es werden nur TraNET FE und das Lokale Device gelistet. Weitere TraNET EPC und PPC Geräte müssen auf den jeweiligen Rechnern direkt kontrolliert werden.





Achten Sie darauf, dass im Dropdown-Menü **"Server"** ausgewählt ist. Im Textfenster finden Sie nun **alle Informationen** zu Ihrem TraNET System.

Ein **Update des TPC-Servers** beinhaltet die entsprechende Version des **TPC-Server** selbst, der **Firmware für die TPCX/TPCE Module** und des **Windows Treibers**.
Ein Update für TranAX beinhaltet lediglich die Bedien-Software TranAX.

Bei Fragen und Problemen speichern Sie die Server-Einstellungen durch klicken des Knopfs "Save" und senden Sie diese Datei anschliessen zusammen mit Ihrer Fehlerbeschreibung an die zuständige Support-Stelle.

Es wird generell empfohlen, die **gesamte Software gemäss Installation CD zu installieren** um allfälligen Inkompatibilitäten zwischen den Software Versionen vorzubeugen.

30.3.1 Beispiel Windows XP

```
Server number 0 started
============================================================
Operation System: Microsoft Windows XP
Server version: 10305

============================================================
Board number 0 found
Driver Version 2.3.3 found
I2C Delay Factor: 5
Firmware "Version A.1.6.2 Overload Reset changed" found
Starhub Version 5 found
```

Hier ist ein Windows XP System im Einsatz mit TPC**-Server Verison 1.3.5** (hier als 10305 dargestellt). Die installierten **Treiber** haben die **Version 2.3.3**, die **Firmware** des TPCX-Moduls die **Version 1.6.2**. Der **Sternverteiler** (Starhub) hat die **Verison 5**.

30.3.2 Beispiel Windows 7

```
Server number 0 started
============================================================
Operation System: Microsoft Windows 7
Server version: 10305

============================================================
Board number 0 found
Driver Version 2.3.3 found
I2C Delay Factor: 4
Firmware "Version A.1.6.2 Overload Reset changed" found
No Synchronisation found
```

In diesem Fall ist ein Windows 7 System im Einsatz mit derselben Software wie oben beim Beispiel mit Windows XP. Die Meldung **"No Synchronisation found"** deutet darauf hin, dass entweder ist **kein Sternverteiler vorhanden** ist, dieser **nicht richtig angeschlossen** ist oder **defekt ist**.

30.3.3 TraNET FE

```
Server number 0 started
============================================================
Model Type: TraNET-FE
Operation System: Linux
Server version: 10305

============================================================
Board number 0 found
Driver Version 2.3.2 found
I2C Delay Factor: 10
Firmware "Version A.1.6.2 Overload Reset changed" found
Starhub Version 1 found
```

Hier steht oben zusätzlich **Model Type: TraNET FE**. Als Betriebssystem kommt hier in **Linux** zum Einsatz. Die Übrigen Informationen sind identisch, ausser dass hier **Sternverteiler** die Firmware **Version 1** hat.

30.4 Fehlermeldungen

30.4.1 TranAX

In TranAX können die folgenden Fehlermeldungen im Control Panel angezeigt werden:

Meldung	Ursache	Lösung
"Hardware failure"	Kein TPCX/TPCE Modul vorhanden	TPCX/TPXE installieren oder TPC-Server in Demo Mode umschalten.
	TPCX/TPCE- Modul nicht richtig eingebaut	Bitte kontrollieren Sie, dass die PCI/PCIe Karte richtig eingebaut und fest im Steckplatz eingebaut ist. Der Rechner muss dazu ausgeschaltet werden!
	Treiber nicht richtig installiert	TPC-Server erneut installieren. Dieser wird mit der CD im Handbuch mitgeliefert.
	PCI/PCIe Steckplatz defekt	Kontrollieren Sie, ab durch einen Wechsel des PCI/PCIe Steckplatzes der Fehler immer noch auftaucht.
	TPCX/TPCE -Modul defekt	Falls keine der oberen Möglichkeiten hilft, muss das TPCX/TPCE-Modul zur genaueren Analyse eingesendet werden.
"Network Error"	Netzwerkkabel nicht angeschlossen	Netzwerkkabel anschliessen, je nach Verbindung muss ein gekreuztes verwendet werden
	TraNET FE oder EPC nicht gestartet	Am Strom anschliessen und starten, wird durch die grün blinkende LED an der Front signalisiert
	Falsche IP Adresse	Kontrollieren, ob im TranAX auch das richtige Gerät oder die lokale Adresse angewählt wurde.
	Falscher Port	Kontrolle ob der richtige Port eingestellt wurde. TranAX muss auf den Port verbinden, der das TraNET zur Verfügung stellt, Kontrolle im TraNET Config.
	Netzwerk Kollisionen	Sicherstellen, dass jedes Gerät eine eigene IP-Adresse besitzt die von keinem anderen Gerät im lokalen Netzwerk verwendet wird. Eventuell den internen IT-Support kontaktieren, dass eine Adresse zugewiesen wird.

 © Elsys AG

30.4.2 TraNET Config Logfile

Weitere wichtige Informationen zum Ihrem Messystem können Sie im Logfile des Servers von TraNET Config entnehmen.

Meldung	Ursache	Lösung
"No Synchronisation found"	Es ist kein Sternverteiler eingebaut	Kein Fehler
	Sternverteiler nicht angeschlossen	Kontrolle der Verbindungskabel zwischen Sternverteiler und TPCX/TPCE-Modulen.
	Sternverteiler defekt	Falls keine der oberen Möglichkeiten hilft, muss der Sternverteile zur genaueren Analyse eingesendet werden.
Keine Einträge vorhanden *"Board number x found"*	Kein TPCX/TPCE Modul vorhanden	Modul erwerben oder TPC-Server in Demo Mode umschalten.
	TPCX/TPCE - Modul nicht richtig eingebaut	Bitte kontrollieren Sie, dass die PCI/PCIe Karte richtig eingebaut und fest im Steckplatz eingebaut ist. Der Rechner muss dazu ausgeschaltet werden!
	Treiber nicht richtig installiert	TPC-Server erneut installieren. Dieser wird mit der CD im Handbuch mitgeliefert.
	PCI/PCIe Steckplatz defekt	Kontrollieren Sie, ab durch einen Wechsel des PCI/PCIe Steckplatzes der Fehler immer noch auftaucht.
	TPCX/TPCE -Modul defekt	Falls keine der oberen Möglichkeiten hilft, muss das TPCX/TPCE -Modul zur genaueren Analyse eingesendet werden.
Keine Einträge vorhanden *"Server number 0 started" fehlt*	TPC-Server nicht installiert oder fehlerhafte Installation	Installieren Sie TPC-Server mit der mitgelieferten CD neu, Eventuell ist ein Neustart des Systems notwendig.

Dokumentinformationen

Stand TranAX	3.4.1.1200
Dateiname:	TranAX_3.4.1_Manual_DE.docx
Status:	Freigegeben

Letzte Änderungen

Datum:	01.09.2014
Autor:	KL/TBE

© Elsys AG